Actuate e.Reporting Suite 5

Building an e.Reporting Web Site

Release 5

Contains information proprietary to:

Actuate Corporation
701 Gateway Boulevard
South San Francisco, CA 94080
http://www.actuate.com

Document No. 010930-2-630301 September 25, 2001

Contents

Chapter 4
ReportCast directives 75

Introduction

About Actuate e.Reporting Suite 5

Actuate is the leading provider of information delivery solutions for e.Business. e.Business customers use Actuate® e.Reporting Suite 5 to develop and deploy high resolution structured content to hundreds of thousands of users. Actuate takes web reporting to the next level by providing options for needs as varied as seamless personalized web pages and traditional online and printed reports.

Actuate's customer list includes commercial banks, securities, financial services, insurance, high tech, telecom, .com, internet, global 2000, and federal government. OEMs, system integrators, and others building e.Business sites for information delivery face challenges. Actuate e.Reporting Suite 5 offers the following solutions.

Challenge	**Actuate solution**
Delivering high resolution information	DHTML provides a fast, no-download option
Viewing structured content	Supporting standard browsers means there is no need to support installations of plug-ins for hundreds of thousands of users
Compromising information display because of lack of integrated tools	Provides template-based design and display
Exploding use of web-based content delivery	Ability to support a million hits per day per CPU
Delivering personalized secure information	Open security directory integration and page security

Challenge	Actuate solution
Reusing existing integrated content	Open server provides access to content from other applications
Maintaining data integrity on hard copy	PDF provides high-resolution printed copy
Transferring information into other applications	XML output provides access to data across applications

Actuate tools and reports do the following:

- Solve complex data access problems.
- Solve formatting problems that go beyond the scope of other tools.
- Scale to support hundreds of thousands of users.

The following summary describes the products in Actuate e.Reporting Suite 5.

Product name	Use
Actuate e.Report Designer Professional	An object-oriented application used by professional developers of structured content to design, build, and distribute report object designs for delivery on the Web. The Actuate Basic Language and Actuate Foundation Class Library support extensive customization capabilities.
Actuate SDK (Software Development Kit) included as part of Actuate e.Report Designer Professional	Actuate ActiveX Controls embed Actuate reporting functionality into custom applications. Actuate Requester API accesses attributes and values of report parameters, changes the values of report parameters, controls how and when a report is generated, displays and prints reports, and configures report print setup. Access the Requester API using Actuate Basic, Visual Basic, C, or C++. Actuate search extension API supports developing search extensions to transfer data to any third-party productivity or analysis tool.

Product name	Use
	Actuate report server API implements common Report Encyclopedia tasks, integrates report server features into existing corporate applications, automates routine or time-consuming tasks, and implements new feature groupings for custom business processes. Access the report server API using C++. Actuate Report Server Security Extension supports the use of third-party security tools. Actuate archive driver supports the use of third-party archiving software and hardware.
Actuate e.Report Designer	An application that complements e.Report Designer Professional and is used by business users to design and distribute a variety of reports. These reports require no programming. This application supports both modifying complex reports and using sophisticated components from libraries.
Actuate e.Report Designer Java™ Edition	A report development application used by Java developers to design and distribute a variety of reports. 100% Java compliant, the e.Report Designer Java Edition includes both AWT and Swing APIs.
Actuate e.Spreadsheet Designer	An application that supports designing, creating, and distributing custom spreadsheets over the web. Users can dynamically generate richly formatted Excel and spreadsheet-based reports from Actuate e.Reporting Server. These spreadsheets can be part of an application, an applet, or a Java Bean.

Product name	Use
Actuate e.Reporting Server	A server application that generates Live Report Documents, manages them in the Report Encyclopedia®, and makes them available to users. This application includes open server functionality that supports the use of third-party report generators with the Actuate e.Reporting Server. This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers. This product also includes Actuate ReportCast™ that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites.
Actuate e.Reporting Server Progress Edition	A server application designed to work exclusively with Progress databases that generates Live Report Documents, manages them in the Report Encyclopedia®, and makes them available to users. This application includes open server functionality that supports the use of third-party report generators with the Actuate e.Reporting Server. This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers. This product also includes Actuate ReportCast that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites.
Actuate Advanced e.Reporting Server	An application that adds page security to the basic e.Reporting Server. Page security supports personalized viewing of parts of reports for various users.

Product name	**Use**
Actuate Advanced e.Reporting Server Progress Edition	An application designed to work exclusively with Progress databases that adds page security to the basic e.Reporting Server. Page security supports personalized viewing of parts of reports for various users.
Actuate e.Analysis	An application used to transform data from an Actuate e.report into interactive information. Users can view and analyze data to determine relationships and trends. Actuate e.Analysis is an optional product that installs with ReportCast and the e.Reporting Server and extends its functionality.
Actuate e.Spreadsheet Option	An open server application that generates Excel spreadsheets from e.Spreadsheet Designer files. With this product customers manage spreadsheet reports and analytics within the Actuate e.Reporting Server, and save Actuate reports as richly formatted Excel spreadsheets.
Actuate End User Desktop	An application used by end users to request, generate, view, and print report documents. The ReportQuery™ capabilities enable seamless transfer of data from an Actuate report to any productivity tool or analysis tool.
Actuate Viewer	Application for end users to find, view, and print report documents. The ReportQuery capabilities are also part of the Actuate Viewer.
Actuate Live Report Extension (LRX)	Application for end users that works with both Microsoft Internet Explorer and Netscape Navigator to support report viewing and printing on the Web.

Actuate Viewer and Actuate Live Report Extension (LRX) are included with all products except Actuate e.Report Designer Java Edition and Actuate e.Spreadsheet Designer.

About *Building an e.Reporting Web Site*

Building an e.Reporting Web Site is a guide to providing access to the Actuate Report Encyclopedia on the World Wide Web.

Building an e.Reporting Web Site includes the following chapters:

- *Introduction*. This chapter provides an overview of this guide, the Actuate e.Reporting Suite, and the typographical conventions used.
- *Chapter 1. Using ReportCast.* This chapter introduces the e.reporting web site, and explains what you need to deploy, navigate, and access e.reports over the Web. This chapter describes how to work with ReportCast channels.
- *Chapter 2. Administering ReportCast.* This chapter describes how to administer ReportCast. This chapter explains how to start and stop ReportCast, configuration tasks, using CGI with ReportCast, and tuning ReportCast performance.
- *Chapter 3. Working with ReportCast templates.* This chapter describes ReportCast templates, and explains how to create custom templates.
- *Chapter 4. ReportCast directives.* This chapter describes the ReportCast directives.
- *Chapter 5. ReportCast scripting language reference.* This chapter describes ReportCast template scripting language and ReportCast scripting language variables.
- *Chapter 6. Internal ReportCast security.* This chapter describes the ReportCast user authentication and security facilities.
- *Chapter 7. Security for Internet deployment.* This chapter describes security issues for deploying e.reporting web sites over the internet.

About Actuate e.Reporting Server product

Actuate e.Reporting Server documentation includes printed manuals, installation guides, online help, user documentation in PDF and HTML format, and release notes. Information about the product that could not be included before the book printing deadline is in the release notes.

The Actuate web site, http://www.actuate.com, contains late-breaking news about the product and its features, as well as product update information. To obtain the password necessary to access the portion of the web site available only to customers, telephone Actuate Customer Support. The engineers in Actuate Customer Support can also help you with technical questions about the product according to your service contract. The Customer Support telephone number, fax number, and e-mail information can be found among the printed materials in the product box.

The printed documentation includes the following manuals.

For information about	**See the following**
Installation	Installation guides
Late-breaking information about the software and documentation	Release notes
Users, groups, privileges, and roles	*Administering the Report Encyclopedia*
Printers and printing requests	
Process groups	
Building and managing your first Report Encyclopedia	
Viewing, running, finding, and printing reports	*Using e.Reports*
Overview of the Actuate report server architecture	*Actuate e.Reporting Server Guide*
Report server administration	
Database connections	
Report Encyclopedia utilities	

For information about	See the following
Actuate ReportCast, its templates, URL directives, scripting language, security and user authentication facilities.	*Building an e.Reporting Web Site*
Terminology map Glossary	*Actuate Glossary*

Online documentation

The information in the printed manuals is also available as online books in Adobe Acrobat PDF format and in the online help system for the Actuate products. For products without a Windows interface, such as the various versions of Actuate e.Reporting Server, Actuate e.Report Designer Java Edition, Actuate ReportCast, and Actuate e.Analysis, we provide HTML help files. In addition Actuate e.Spreadsheet Designer provides Java online help files. The HTML and Java help files install automatically with the product. These files can be viewed with standard browsers.

Using online manuals

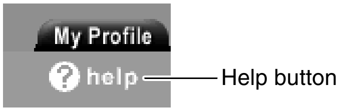
The online manuals install automatically with the product. On the product CD, you can also find those files in the Manuals directory. Open the introductory PDF file to get an overview of the manuals. The items in the table of contents and the page numbers in the index both contain links to the appropriate topics in the text. In the index, you access the link by positioning your cursor over the page number, not the topic.

Online help

Actuate products provide both context-sensitive online help about the product and report-specific online help about the report you are viewing. Actuate e.Reporting Suite makes it possible for developers to create customized report-specific online help.

Using online help

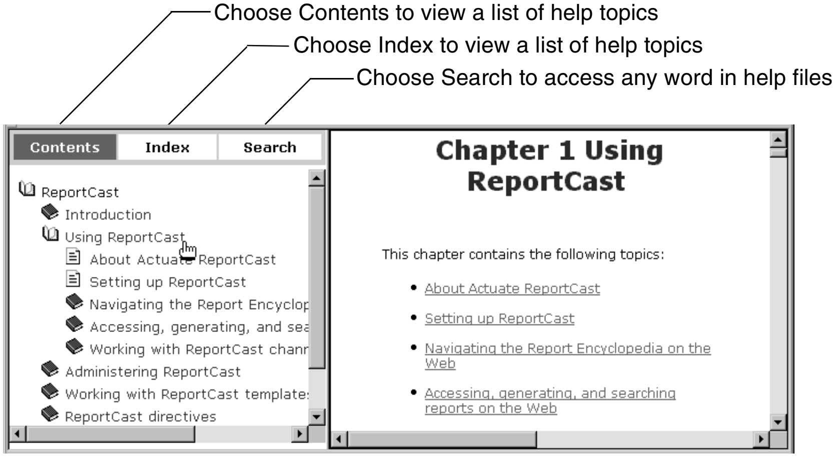
To access online help, choose Help in the toolbar as shown in the following illustration.



Choose a link to access the online help files, or under PDF files, choose a link to the Adobe Acrobat PDF files.

Using the Actuate online help system

Use two windows to access and view information in the e.Reporting Suite help system. The window on the left displays the table of contents or the index of the online help system. The window on the right displays the contents of the online help topics.


Choose Contents to view a list of help topics
Choose Index to view a list of help topics
Choose Search to access any word in help files
Contents
Index
Search

- ReportCast
  - Introduction
  - Using ReportCast
    - About Actuate ReportCast
    - Setting up ReportCast
    - Navigating the Report Encyclop
    - Accessing, generating, and sea
    - Working with ReportCast chanr
  - Administering ReportCast
  - Working with ReportCast template:
  - ReportCast directives

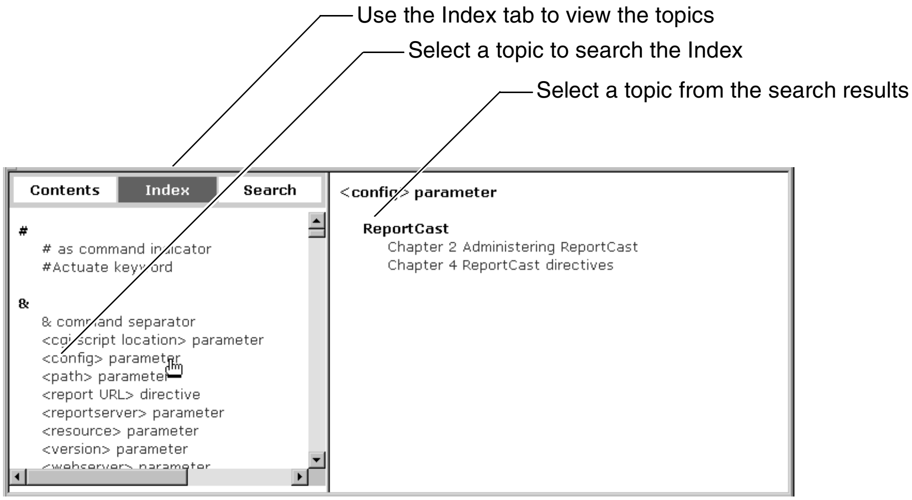
Chapter 1 Using ReportCast

This chapter contains the following topics:

- About Actuate ReportCast
- Setting up ReportCast
- Navigating the Report Encyclopedia on the Web
- Accessing, generating, and searching reports on the Web

The tabs at the top of the left window access different views. Use these tabs to switch views among the Table of Contents, Index, and Search. The Table of Contents provides an overview of the help file contents. For example, in the preceding window, you see the introduction to the documentation for the e.Reporting Server product. The Search tab accesses any word in the help files.

The following two illustrations show an example of the Index and the result of an Index search.The following illustration shows the result of the search as it appears in the window on the right.


Use the Index tab to view the topics
Select a topic to search the Index
Select a topic from the search results
Contents
Index
Search

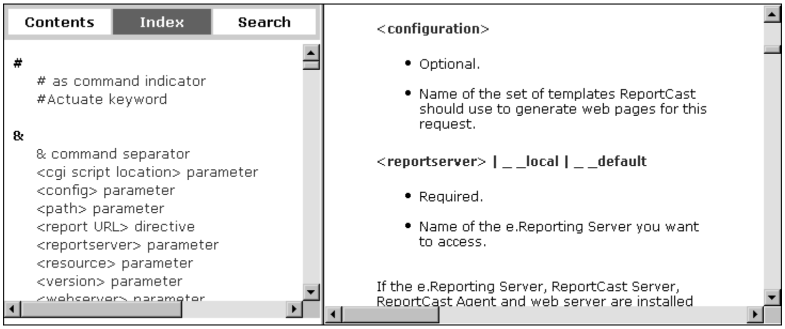
#
as command indicator
#Actuate keyword

&
& command separator
<cgi script location> parameter
<config> parameter
<path> parameter
<report URL> directive
<reportserver> parameter
<resource> parameter
<version> parameter
<webserver> parameter

<config> parameter

ReportCast
Chapter 2 Administering ReportCast
Chapter 4 ReportCast directives

To view the topic double-click the topic in the list. The topic displays in the window on the right.



#
as command indicator
#Actuate keyword

&
& command separator
<cgi script location> parameter
<config> parameter
<path> parameter
<report URL> directive
<reportserver> parameter
<resource> parameter
<version> parameter
<webserver> parameter

<configuration>

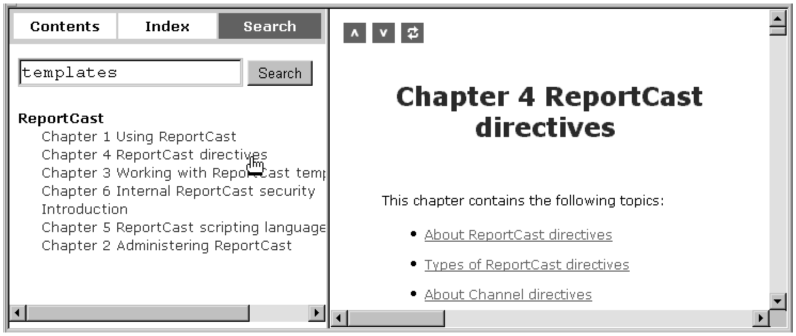
- Optional.
- Name of the set of templates ReportCast should use to generate web pages for this request.

<reportserver> | __local | __default

- Required.
- Name of the e.Reporting Server you want to access.

If the e.Reporting Server, ReportCast Server, ReportCast Agent and web server are installed

Use the Search tab to find all instances of a keyword in the help files. The following illustration shows the results of a search for the keyword, templates. The topics containing the keyword appear under Search. Double-click a topic in the Search list to display the topic in the right window.



ReportCast
Chapter 1 Using ReportCast
Chapter 4 ReportCast directives
Chapter 3 Working with ReportCast temp
Chapter 6 Internal ReportCast security
Introduction
Chapter 5 ReportCast scripting language
Chapter 2 Administering ReportCast

Chapter 4 ReportCast directives

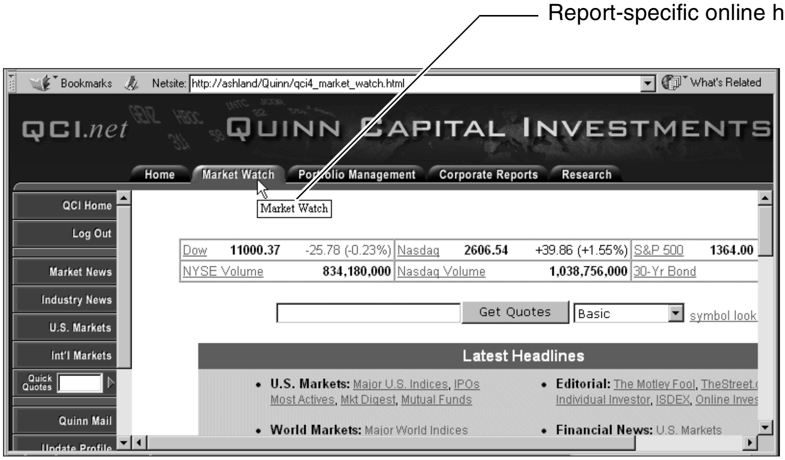
This chapter contains the following topics:

- About ReportCast directives
- Types of ReportCast directives
- About Channel directives

Using report-specific online help

During the design phase, report developers have the option to include report-specific online help. For example, the report developer can add comments to give further detail about specific report objects or to explain how calculations were created.

Report-specific online help



Dow	11000.37	-25.78 (-0.23%)	Nasdaq	2606.54	+39.86 (+1.55%)	S&P 500	1364.00
NYSE Volume		834,180,000	Nasdaq Volume		1,038,756,000	30-Yr Bond	

- U.S. Markets: Major U.S. Indices, IPOs Most Actives, Mkt Digest, Mutual Funds
- Editorial: The Motley Fool, TheStreet.c Individual Investor, ISDEX, Online Inves
- World Markets: Major World Indices
- Financial News: U.S. Markets

For detailed information about report-specific online help see Chapter 3, “Viewing a report from the desktop,” in *Using e.Reports*.

Typographical conventions

The following table describes the typographical conventions used in this guide.

Item	Convention	Example
Code examples	Sans serif	Dim As String
File names	Initial letter capitalized. Except e. Report Designer Java Edition, where file names are case sensitive	Detail.roi
Key combination	A + sign between the keys means to press both keys at the same time	Ctrl+Shift
Menu items	Capitalized. No bold.	File
Submenu items	Separated from main menu item with small arrow	File➛New
User input or user response	Sans serif	M*16*
User input in XML and Java code	Italics	chkjava.exe *cab_name*.cab

Syntax conventions

The following table describes the symbols used to present the syntax.

Symbol	Description	Example
[ ]	Optional item	[Alias<alias name>]
	Array subscript	blob[ ]
< >	Argument you must supply	<expression to format>
	Delimiter in XML	<tr>

Symbol	Description	Example
{ }	Groups two or more mutually exclusive options or arguments, when used with a pipe	{While | Until}
	Defines array contents	{0, 1, 2, 3}

	Delimiter of code block	public ACJDesigner( )
		{
		}

|	Separates mutually exclusive options or arguments in a group	Exit {Do | For | Function | Sub}
	Java OR operator	int var |4

Chapter

1

Using ReportCast

This chapter contains the following topics:

- About Actuate ReportCast
- Setting up ReportCast
- Navigating the Report Encyclopedia on the Web
- Accessing, generating, saving, and searching for reports on the Web
- Working with ReportCast channels

About Actuate ReportCast

Actuate ReportCast provides access to the Actuate e.Reporting Server from the World Wide Web. Using ReportCast, you can access and work with reports in the Actuate Report Encyclopedia with any web browser.

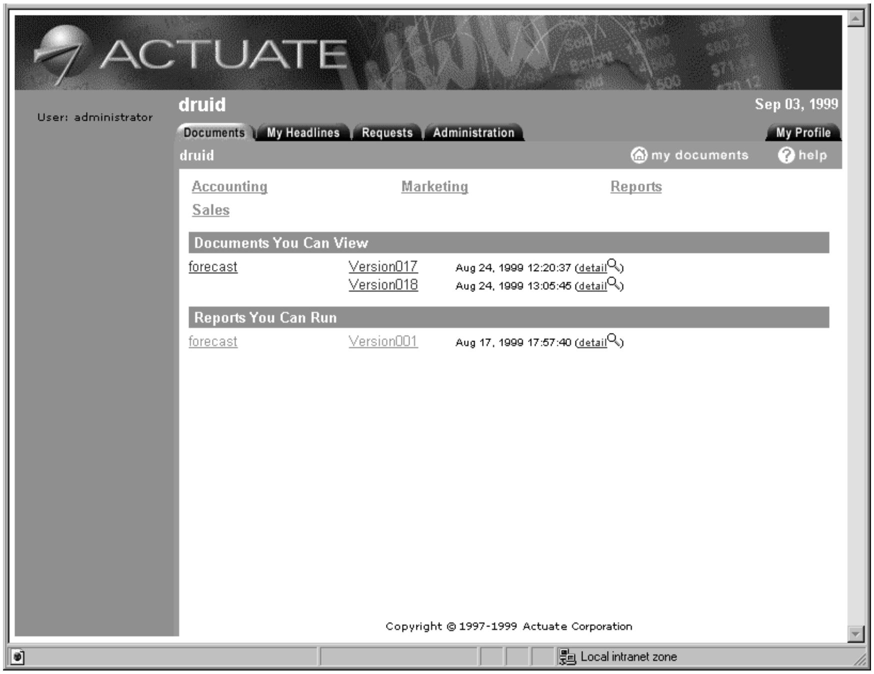
ReportCast technology consists of:

- ReportCast Agent—ReportCast's interface to the web server. ReportCast Agent is a dynamic-link library (DLL) that runs as part of your web server. There is no user interface to the DLL.
- ReportCast Server—ReportCast's interface to the Actuate e.Reporting Server. ReportCast Server is a service and runs as a process separate from the web server and ReportCast Agent. There is no user interface to the ReportCast Server service.

Previously, to view Actuate reports over the Web you used Actuate LRX (Live Report Extension) for Netscape or Microsoft plug-ins installed with your web browser. With ReportCast, you can view e.reports directly using any web browser. The navigation options of the two viewing methods differ slightly.

ReportCast generates HTML pages that are adaptations of the Actuate Navigator interface, then sends those HTML pages to the web browser. For example, an HTML form that accepts parameter values for report generation replaces the Requester dialog box of the Navigator. Using ReportCast technology you can access HTML reports and DHTML reports without needing any additional software installed on your desktop system.

The following illustration shows the default reporting web site.



druid

Sep 03, 1999

druid

Accounting Marketing Reports

Sales

Documents You Can View

forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

The URL format of the default reporting web site is:

http://<webserver>/acweb/<reportserver>

- <web server> is the name of the web server running the web interface to the Report Encyclopedia.
- acweb is a keyword indicating that you want to access the Report Encyclopedia.
- <report server> is the name of the e.Reporting Server with the encyclopedia you want to access.

Users navigate the reporting web site by clicking links on the reporting web page, or by entering ReportCast directives as part of a URL.

ReportCast provides a set of default web page templates that determine the appearance of the pages that display in the browser. Commands can be written in a combination of the ReportCast scripting language, Javascript, and standard HTML tags and are embedded in the template. Webmasters have these options for using the templates:

- Use the default templates
- Customize the default templates
- Create entirely new templates

Setting up ReportCast

As you create structured content from databases, you deploy e.reports over the World Wide Web. A typical setup includes:

- A web server
- An Actuate installation consisting of:
  - e.Reporting Server
  - An e.report development tool such as e.Report Designer Professional or e.Report Designer
  - Administrator Desktop
- ReportCast

A webmaster must have administrator privileges to the web server and report server to effectively manage the e.reporting web site.

Navigating the Report Encyclopedia on the Web

Actuate's web interface to the Report Encyclopedia allows you to access and work with reports on the Report Encyclopedia using a web browser. The interface provides web-based Actuate Navigator functionality similar to that provided by the End User Desktop and Administrator Desktop. The web interface also provides a service called ReportCast channels. ReportCast channels use push technology to provide reports to particular web channels. By subscribing to the channels that interest you, you can easily find and view reports on topics of particular importance to you.

Check with your e.Reporting Server administrator to find out if you can access reports on the Web.

To navigate through the Report Encyclopedia using a web browser, click hyperlinks contained on the Encyclopedia's web pages, or enter special URLs or directives, in the Address or Location window of your web browser. For more information about directives, see Chapter 4, "ReportCast directives."

For more information about navigating the Report Encyclopedia, see Chapter 2, "Working with the desktop Navigator," in *Using e.Reports*, the LRX online help, or the online manual in .PDF format. The illustrations in this chapter use the standard web pages provided by Actuate. Your e.Reporting Server administrator may customize the design of the web pages for your site, so your pages may appear different from those in the illustrations.

Accessing the Report Encyclopedia on the Web

To access your Report Encyclopedia, bring up your web browser and enter a URL similar to the following:

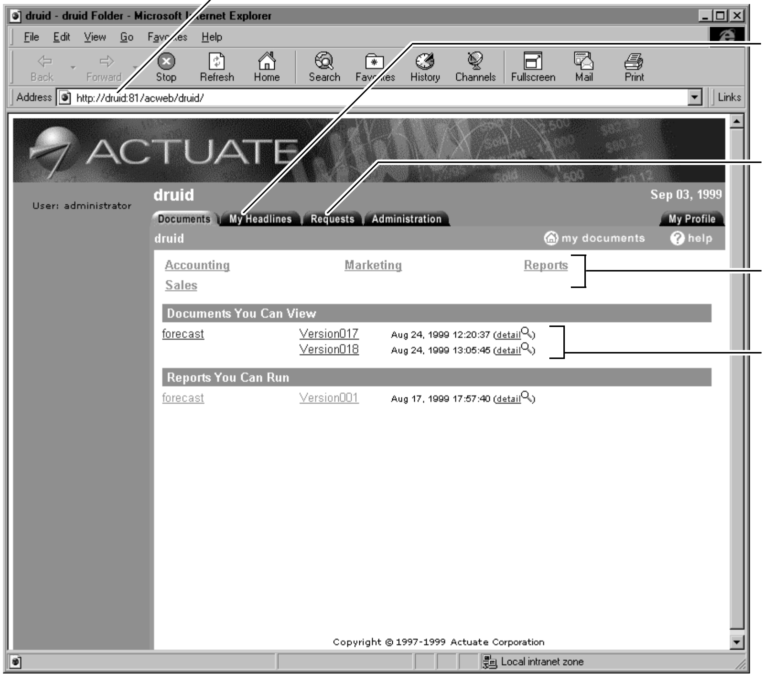
```
http://<web server>/acweb/<report server>
```

- <web server> is the name of the web server running the web interface to the Report Encyclopedia.
- acweb is a keyword indicating that you want to access the Report Encyclopedia.
- <report server> is the name of the e.Reporting Server with the encyclopedia you want to access.

For example, to access reports on an e.Reporting Server called Druid from a web server called Druid, running on port 81:

```
http://druid:81/acweb/druid
```

Depending upon your setup, a Login dialog may appear. If it does, enter the user name and password you normally use to access your Report Encyclopedia. A Report Encyclopedia web page appears in the browser.



Address: http://druid:81/acweb/druid/

User: administrator

druid — Sep 03, 1999

druid

Accounting   Marketing   Reports
Sales

Documents You Can View

forecast   Version017   Aug 24, 1999 12:20:37 (detail)
            Version018   Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast   Version001   Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

URL to access the Druid Report Encyclopedia.

Access your personal channel, subscribe to channels.

Check on active or scheduled requests.

Available Report Encyclopedia folders.

Click these links to get detailed information about a folder.

From this main page, you can:

- View the contents of the Report Encyclopedia.
- Access your personal channel (your completed reports folder).
- Access the available Report Encyclopedia folders.
- Get detailed information about folders.
- Sort the folders by name, type, or creation date and time.
- Request report runs.
- Get status information about requests to run reports.

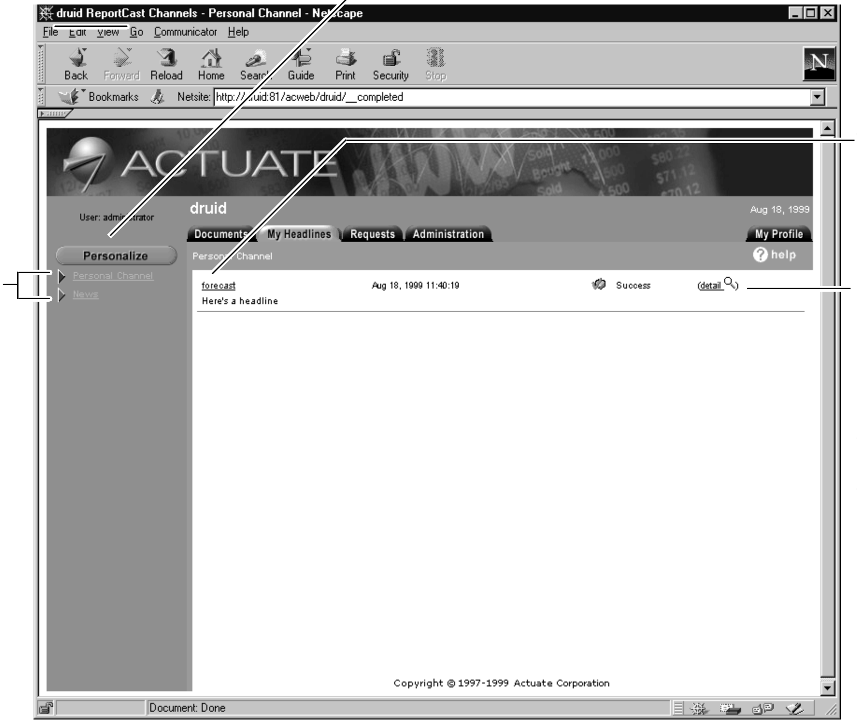
Requests to run reports or to get status information about requests time out after 120 seconds if ReportCast cannot contact the e.Reporting Server.

To ensure that you always see the most current web reporting page, set your web browser to check the date of web pages every time you issue a request to the e.Reporting Server.

To see changes to the Report Encyclopedia that do not involve adding or deleting an item or folder (for example, updating an existing item), force-reload the web reporting page. To force-reload a web reporting page in Netscape, hold the shift-key down at the same time as clicking the reload button. In Microsoft Internet Explorer, clear the cache and then reload the page.

Accessing channels and reports

When you click the My Headlines tab on the main Report Encyclopedia web page, a page such as the following appears in the browser.


druid ReportCast Channels - Personal Channel - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://druid:81/acweb/druid/__completed
ACTUATE
User: administrator
druid
Aug 18, 1999
Documents My Headlines Requests Administration My Profile
Personalize
Personal Channel
help
Personal Channel
News
Document: Done

forecast Aug 18, 1999 11:40:19 Success (detail)
Here's a headline

Copyright © 1997-1999 Actuate Corporation

Subscribe to other ReportCast channels. The Administration tab appears only if you have administrator privileges.

These are the completed reports you can view.

Click these links to access the channels to which you subscribe.

Click the Details icons to examine a report's status, view the report, or delete the status information.

From My Headlines page, you can:

- Click Personalize to subscribe to and unsubscribe from ReportCast channels.
- Click channel icons to access particular channels to which you are subscribed.
- View completed reports.
- Click Details to view a report's status, view the report, or delete status information for the report.

Accessing, generating, saving, and searching for reports on the Web

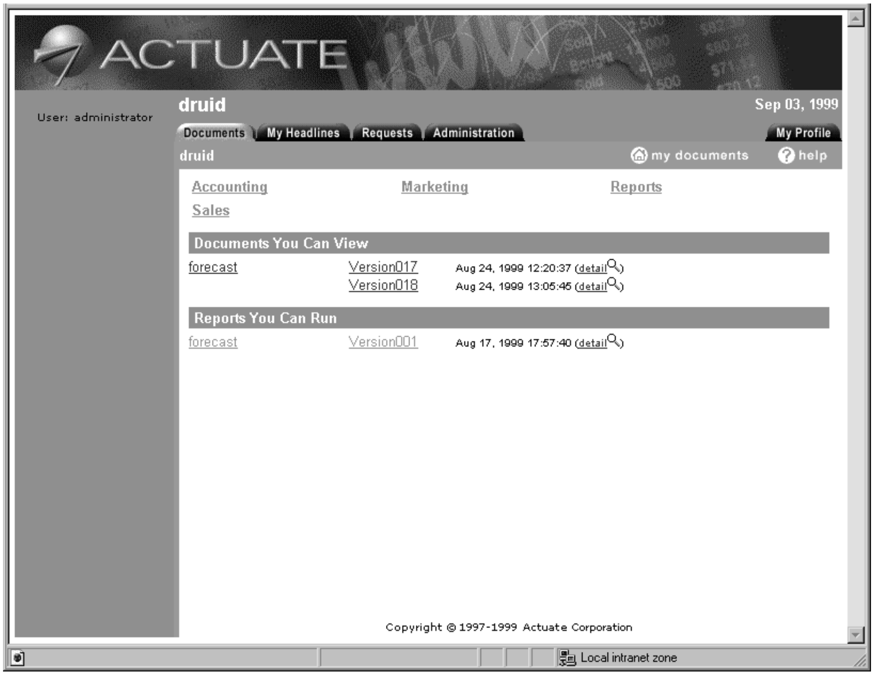
To access and generate reports on the Web:

- Click report executable links in a Report Encyclopedia folder to generate a report.
- Click report links on the My Headlines web page or from other ReportCast channels' pages to view a report.
- Navigate through the Report Encyclopedia folders from the main Report Encyclopedia web page.

Generating a report

When the main web page for the Report Encyclopedia displays, the same list of folders that you normally see in the Administrator Desktop's Navigator window displays. Click a folder's link to display its contents.

Folder contents pages are similar to the following image.



User: administrator

druid

Sep 03, 1999

druid

Accounting Marketing Reports
Sales

Documents You Can View

forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

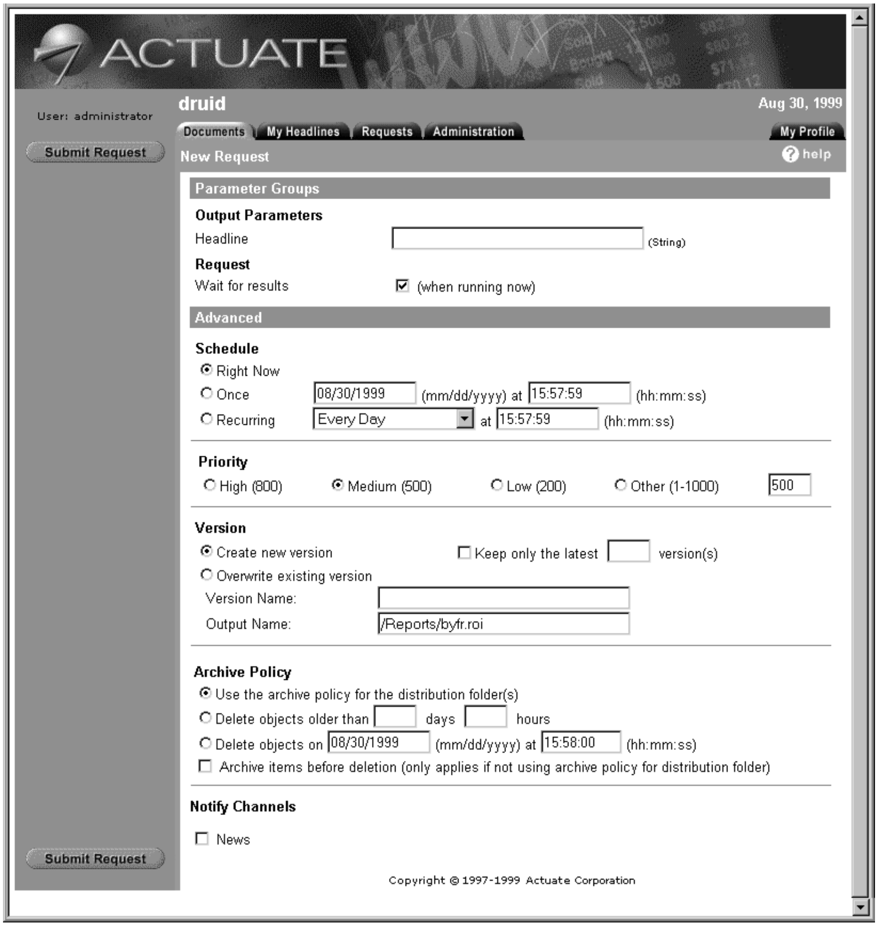
Reports You Can Run

forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

The folder web page lists the folder's contents, including any subfolders, as well as links to your personal folder and to the Active and Scheduled requests web pages.

The difference between this page and your personal channel web page is that the folder contents page lists report executables. To generate a report, click the report executable's link. A requester page similar to the following displays in the browser.



druid

Aug 30, 1999

New Request

Parameter Groups

Output Parameters

Headline (String)

Request

Wait for results ☑ (when running now)

Advanced

Schedule

● Right Now

○ Once 08/30/1999 (mm/dd/yyyy) at 15:57:59 (hh:mm:ss)

○ Recurring Every Day at 15:57:59 (hh:mm:ss)

Priority

○ High (800) ● Medium (500) ○ Low (200) ○ Other (1-1000) 500

Version

● Create new version □ Keep only the latest version(s)

○ Overwrite existing version

Version Name:

Output Name: /Reports/byfr.roi

Archive Policy

● Use the archive policy for the distribution folder(s)

○ Delete objects older than days hours

○ Delete objects on 08/30/1999 (mm/dd/yyyy) at 15:58:00 (hh:mm:ss)

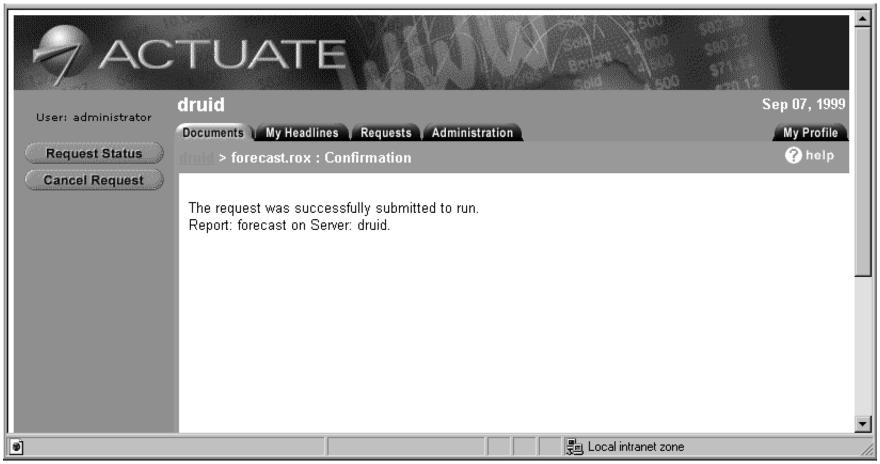
□ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels

□ News

Copyright © 1997-1999 Actuate Corporation

Fill out the requester as you would in the Administrator Desktop, then click Submit Request to submit the request. A confirmation form displays in the browser, similar to the following.



The request was successfully submitted to run.
Report: forecast on Server: druid.

Click Request Status to check on your request's status. To cancel the request, click Cancel Request.

Assigning unique e.report output and version names

You can use the report generation date as a parameter to assign a unique output name and version name to an e.report. Use a date/time expression to specify the report generation date in the Version Name and Output Name fields of the requester form.

Enclose the date/time expression in braces, {}. Build the date/time expression using either Actuate Basic pre-defined date/time formats or Actuate Basic user-defined date/time formats.

The following date/time expression assigns the name Sales report as of the current date. For example, if the current date is March 1, 2001, the name of the e.report is Sales report as of 03-01-2001:

```
Sales report as of {mm-dd-yyyy}
```

ReportCast does not evaluate the date/time expression. The e.Reporting Server evaluates the date/time expression when ReportCast submits the report for generation.

See *Using e.Reports* for more information about specifying a date/time expression as a parameter for an e.report's output name and version name.

Setting report viewing options

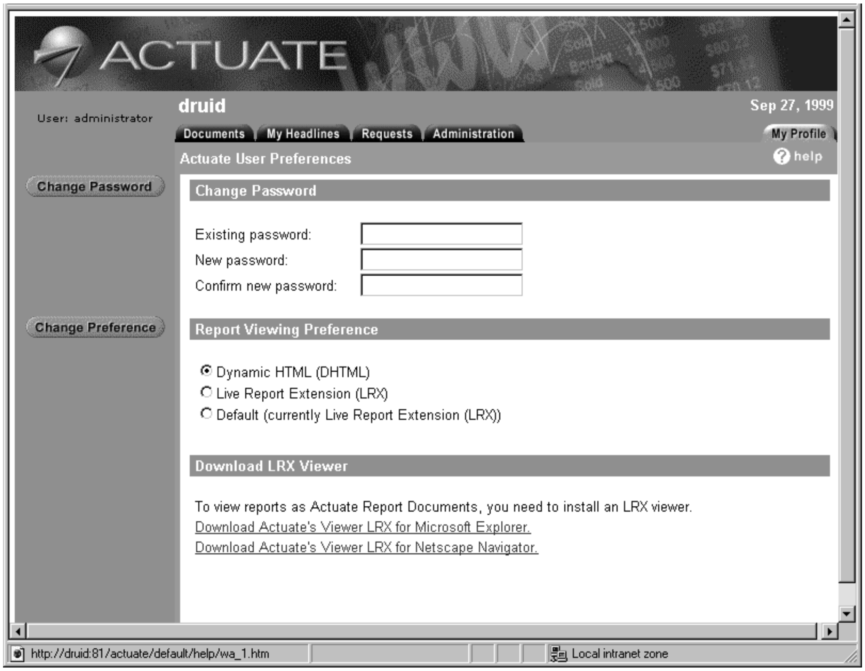
You can tell ReportCast how you want to view reports. Use the User Preferences page to set your report viewing options. Either click Report Viewing Options in the ReportCast page banner or enter a URL similar to the following:

```
http://<web server>/acweb/<e.reporting server>/__requestUserPreferences
```

For example, if your web server’s name is Druid, and your e.Reporting Server also resides on Druid (running on port 81):

```
http://druid:81/acweb/druid/__requestUserPreferences
```

A page similar to the one in the following illustration appears.



Actuate User Preferences

Change Password

Existing password:

New password:

Confirm new password:

Report Viewing Preference

● Dynamic HTML (DHTML)
○ Live Report Extension (LRX)
○ Default (currently Live Report Extension (LRX))

Download LRX Viewer

To view reports as Actuate Report Documents, you need to install an LRX viewer.
Download Actuate's Viewer LRX for Microsoft Explorer.
Download Actuate's Viewer LRX for Netscape Navigator.

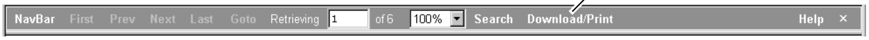
Choose whether to view reports in Dynamic HTML (DHTML) format, or as Actuate reports using the Actuate Live Report Extension (LRX). The default viewing format is DHTML. Your system administrator may have set a system-wide e.Reporting Server default.

You can also download the Actuate Viewer LRX or change your password from the User Preferences page.

For more information about the User Preferences page, see *Using e.Reports.*

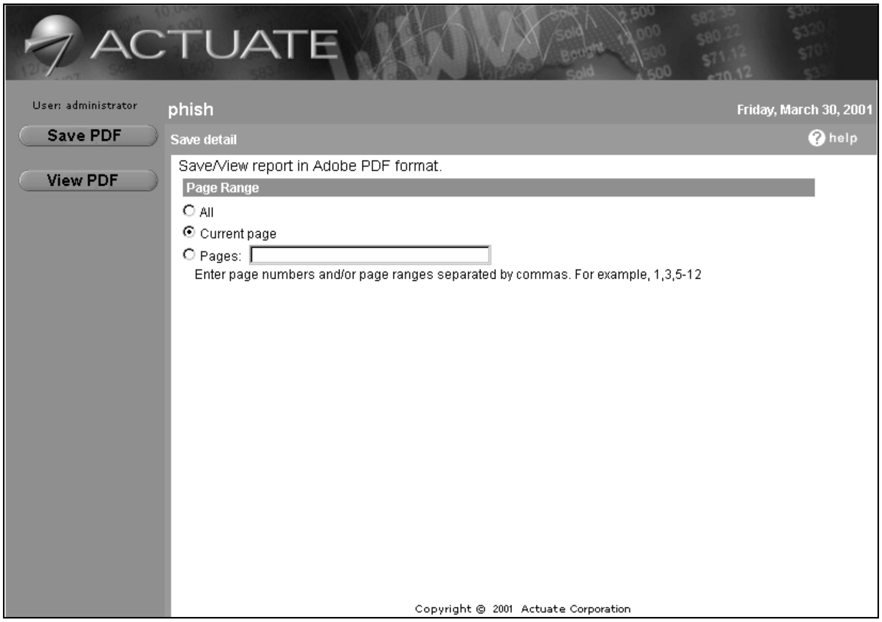
Saving and viewing DHTML reports as PDF

You can save and view DHTML reports in PDF format. From the DHTML report viewing page, choose the Download/Print link.



Download/Print link

The Save detail page appears.



User: administrator

phish

Friday, March 30, 2001

Save detail

Save/View report in Adobe PDF format.

Page Range

○ All

● Current page

○ Pages: [ ]

Enter page numbers and/or page ranges separated by commas. For example, 1,3,5-12

Copyright © 2001 Actuate Corporation

You can specify all or part of the report to save or view in PDF format:

- Click All to save or view the entire report.
- Click Current page to save or view the current page.
- Click Pages: to save or view specific pages or ranges of pages.

To save the DHTML report in PDF format:

1 Choose Save PDF to save the DHTML report in PDF format.

Save As... appears.

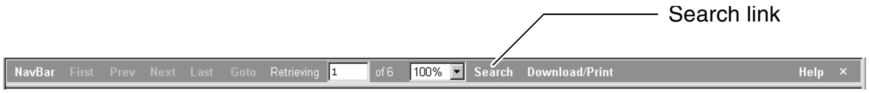
2 Use Save As... to choose the name and location for the PDF format report.

To view the DHTML report in PDF format, choose View PDF. The PDF report displays in the web browser.

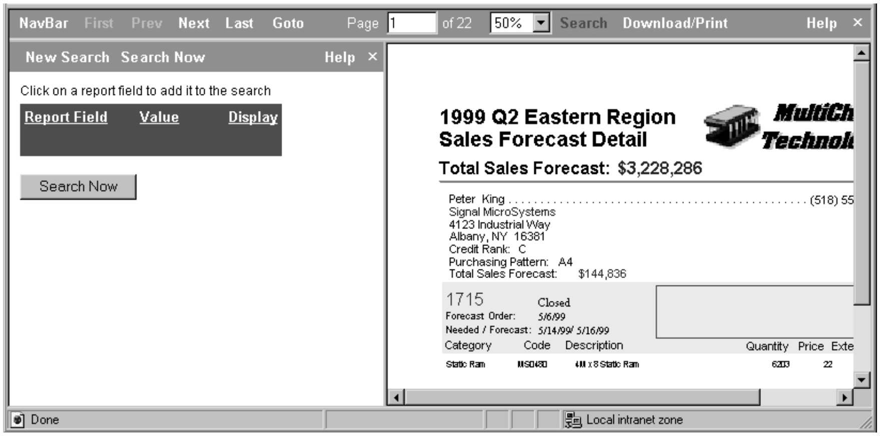
Searching DHTML reports

You can search for information while viewing DHTML reports using the Search Request page. To perform a search:

1 From the DHTML report viewing page, choose Search.



The Search Request page appears.



Click on a report field to add it to the search

1999 Q2 Eastern Region
Sales Forecast Detail

Total Sales Forecast: $3,228,286

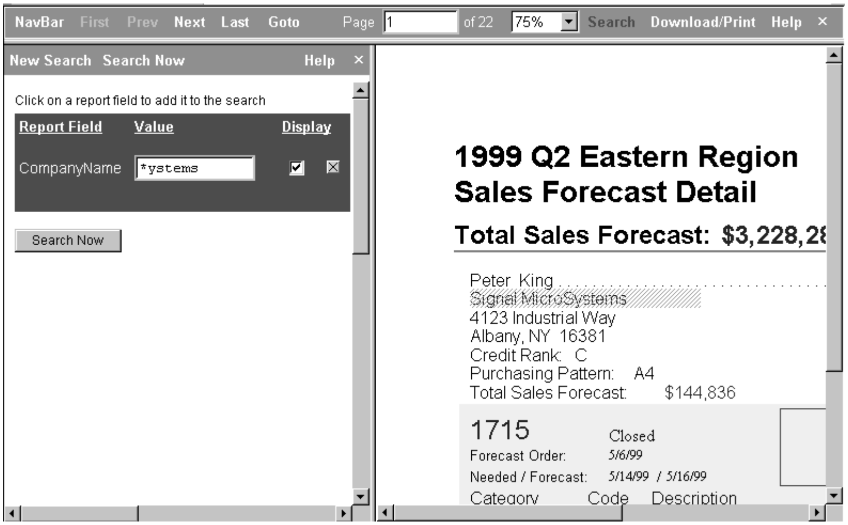
Peter King .... (518) 55
Signal MicroSystems
4123 Industrial Way
Albany, NY 16381
Credit Rank: C
Purchasing Pattern: A4
Total Sales Forecast: $144,836

1715 Closed
Forecast Order: 5/6/99
Needed / Forecast: 5/14/99/ 5/16/99

Category	Code	Description	Quantity	Price	Exte
Static Ram	MS0480	4M x 8 Static Ram	6203	22	

2 Select fields to add to your search criteria. The selected fields appear in the Search List.

3 Type the expression or value to search for. The Search Request page looks similar to the following image.



Click on a report field to add it to the search

Report Field	Value	Display	
CompanyName	*ystems	☑	☒

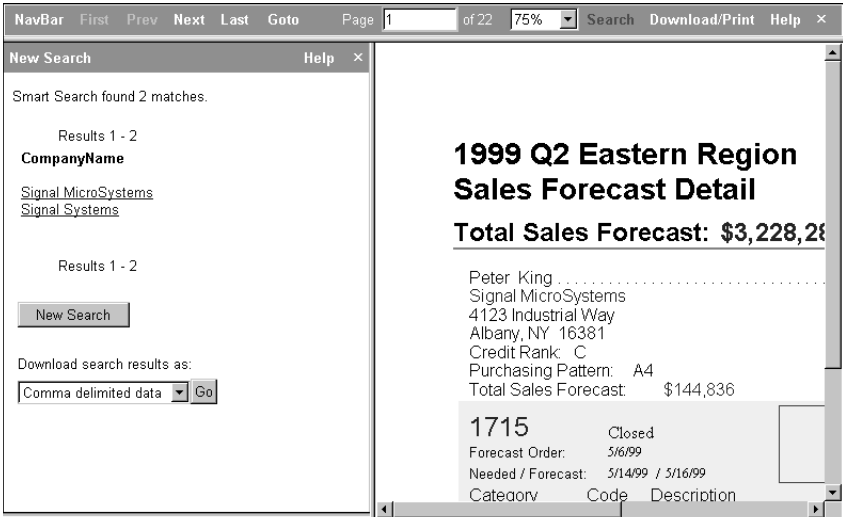
1999 Q2 Eastern Region Sales Forecast Detail

Total Sales Forecast: $3,228,28

Peter King
Signal MicroSystems
4123 Industrial Way
Albany, NY 16381
Credit Rank: C
Purchasing Pattern: A4
Total Sales Forecast: $144,836

1715 Closed
Forecast Order: 5/6/99
Needed / Forecast: 5/14/99 / 5/16/99
Category Code Description

4 Click Search Now. Report query result appears.



Smart Search found 2 matches.

Results 1 - 2

CompanyName

Signal MicroSystems
Signal Systems

Results 1 - 2

Download search results as:

1999 Q2 Eastern Region Sales Forecast Detail

Total Sales Forecast: $3,228,28

Peter King
Signal MicroSystems
4123 Industrial Way
Albany, NY 16381
Credit Rank: C
Purchasing Pattern: A4
Total Sales Forecast: $144,836

1715 Closed
Forecast Order: 5/6/99
Needed / Forecast: 5/14/99 / 5/16/99
Category Code Description

5 Click search result links to examine specific search results.

To remove search criteria, you can do one of two things:

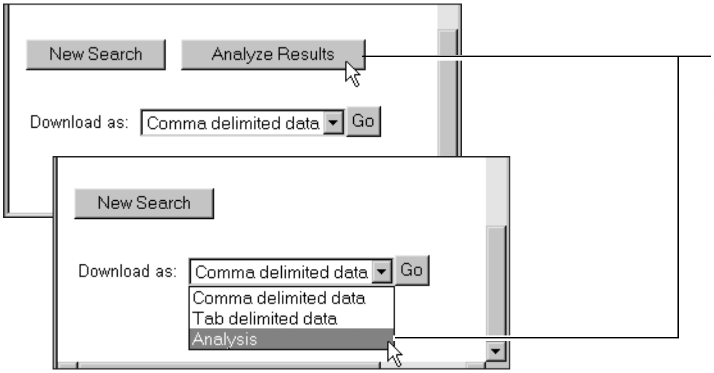
- Click a selected field to de-select it
- Click the X icon next to the selected field

For more information about searching DHTML reports on the Web, see *Using e.Reports*.

Showing and hiding the e.Analysis button

e.Analysis is an add-on application that works with DHTML reports on the Web. You use e.Analysis to analyze data fields selected in a Smart Search. For more information, see *Using e.Analysis*.

If e.Analysis is installed, you launch it by choosing Analyze Results or choosing Analysis on a drop-down menu in the Smart Search results pane, as shown in the following illustration.



Launch e.Analysis by choosing Analyze Results or by choosing Analysis on the drop-down list.

Webmasters can customize how users launch e.Analysis. If e.Analysis is installed and neither of these options is available, check with your webmaster to find out how to launch e.Analysis.

Navigating through the Report Encyclopedia

To navigate the report from the main Report Encyclopedia web page, click a folder’s link. The folder contents page appears in the browser. You can continue clicking subfolder links to display their contents, click a report executable to generate a report, or click report .roi, .rov, or .row links to view reports.

Working with ReportCast channels

ReportCast channels provide easy access to particular reports of interest on the Web. When you subscribe to a channel, its link appears in your personal folder. Click the link to check the channel's contents and view reports as you wish.

Subscribing to a ReportCast channel

To subscribe to a ReportCast channel:

1. Choose My Headlines. Your personal folder page appears.
2. Click Personalize. The channel subscription page appears.

   The subscription page contains a list of available channels, with checkboxes to subscribe to them. Select the checkboxes for the channels to which you wish to subscribe. Click Details to find out more about the channel.
3. Click Apply. A confirmation page similar to the subscription page appears in your browser.
4. Return to your personal channel. Reload the page. The channels you just subscribed to appear on your personal channel web page.

Unsubscribing from a ReportCast channel

To unsubscribe from a ReportCast channel:

1. Choose My Headlines. Your personal folder page appears.
2. Click Personalize. The channel subscription page appears.

   The subscription page contains a list of available channels, with checkboxes. A checked checkbox indicates that you subscribe to a particular channel. Select the checkboxes for the channels from which you wish to unsubscribe. The check is removed.
3. Click Apply. A confirmation page similar to the subscription page displays in your browser.
4. Return to your personal channel. Reload the page. The channels from which you just unsubscribed no longer appear on your personal channel web page.

Chapter

2

Administering ReportCast

This chapter contains the following topics:

- About ReportCast administration
- About ReportCast installations
- Starting and stopping ReportCast
- Setting ReportCast default e.Reporting Server
- Changing the Windows servers ReportCast TCP/IP port
- Changing the ReportCast server machine
- Using non-default web server ports
- Configuring ReportCast port numbers
- Configuring PDF viewing and saving options
- Creating custom configurations
- Mapping custom configurations to locales
- Accessing UNIX ReportCast with CGI scripts
- Working with reports on Windows web sites
- Using CGI scripts for security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning ReportCast performance
- Viewing files using the native web server file system

About ReportCast administration

Administering ReportCast requires:

- Web server and report server administrator privileges
- Familiarity with web server and report server administration tasks
- Familiarity with ReportCast installations

ReportCast administration tasks include:

- Starting and stopping ReportCast
- Setting the default e.Reporting Server
- Changing the Windows server ReportCast TCP/IP port
- Changing the Windows server ReportCast Server machine
- Configuring communication port numbers
- Configuring the ReportCast Server host name
- Creating custom configurations
- Mapping configurations to system-defined language locales
- Accessing ReportCast using CGI scripts
- Working with reports on Windows sites
- Using CGI scripts for ReportCast security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning performance
- Viewing reports using the native web server file system

About ReportCast installations

ReportCast consists of the following components:

- ReportCast Agent handles the interface to the web server. ReportCast Agent is a dynamic-link library (DLL) that runs as part of your web server.
- ReportCast Server handles the interface to the Actuate e.Reporting Server. ReportCast Server is a service and runs as a process separate from the web server and ReportCast Agent.

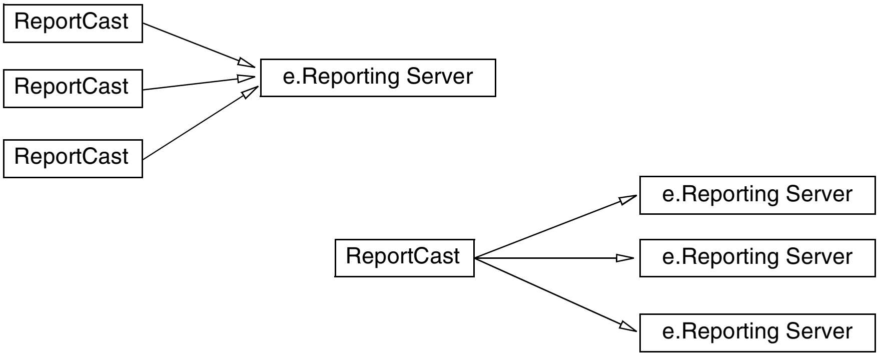
There is no user interface to ReportCast. There is no obvious result of installation.

Windows NT and Windows 2000 support one instance of a ReportCast Server per system. UNIX supports more than one instance of a ReportCast Server per system, as long as you configure each instance to listen on a different port. See Chapter 7, "Security for Internet deployment," for more information about configuring ReportCast Server and e.Reporting Server ports.

A single ReportCast Server can access multiple e.Reporting Servers. For example, you could use a single ReportCast Server accessing multiple e.Reporting Servers if you have a single web server hosting several applications, with each application using its own e.Reporting Server.

Multiple ReportCast servers can access a single e.Reporting Server. For example, you could use a multiple ReportCast Servers accessing one e.Reporting Server configuration if you have several web servers accessing the same e.Reporting Server resource.

The following diagram illustrates two possible ReportCast configurations.



Starting and stopping ReportCast

This section describes how to start and stop ReportCast on Windows servers and UNIX systems.

Starting and stopping ReportCast on Windows server systems

The Windows NT and Windows 2000 ReportCast Agent operates as part of the web server. ReportCast Agent starts and stops automatically when you start and stop the web server.

The Windows NT and Windows 2000 ReportCast Server is a service. To stop the Windows NT and Windows 2000 ReportCast Server service:

1 Start➛Settings➛Control Panel.

  Control Panel appears.

2 Double-click Services.

  Services appears.

3 Click Actuate ReportCast Server 5.

4 Click Stop.

  The ReportCast Server service stops.

To start the Windows NT and Windows 2000 ReportCast Server service:

1 Start➛Settings➛Control Panel.

  Control Panel appears.

2 Double-click Services.

  Services appears.

3 Click Actuate ReportCast Server 5.

4 Click Start.

  The ReportCast Server service starts.

Starting and stopping ReportCast on UNIX systems

UNIX ReportCast has these components:

- NSAPI extension or CGI component
- ReportCast service

The NSAPI or CGI components start automatically when the web server starts, or when a user navigates to a web reporting site URL. These components stop automatically when the web server stops.

The ReportCast service starts and stops separately.

Starting a UNIX ReportCast service

The ReportCast installation adds the ReportCast service startup script to the actuate/bin directory in the netscape/plugins directory for the web server that runs ReportCast. To start the ReportCast service, run the following script:

```
<web server directory>/startreportcast.sh
```

- <web server directory> is the netscape/plugins/actuate/bin directory path for the web server that runs ReportCast.
- startreportcast.sh is the ReportCast service startup script.

Stopping a UNIX ReportCast server

To stop the UNIX ReportCast server, run the following script:

```
<web server directory>/stopreportcast.sh
```

- <web server directory> is the netscape/plugins/actuate/bin directory path for the web server that runs ReportCast.
- stopreportcast.sh is the ReportCast server shutdown script.

Installation process for Sun Solaris and IBM-AIX

The ReportCast installation process for the Sun Solaris and IBM-AIX ReportCast service adds the service startup script to the Netscape web server startup script. Sun Solaris and IBM-AIX ReportCast services start when the web server starts.

Setting ReportCast default e.Reporting Server

You can define a default e.Reporting Server for ReportCast. This allows you to make use of the _ _ default e.Reporting Server specification in ReportCast directives. You use the registry key on Windows NT and Windows 2000 or environment variable on UNIX AC_REPORTCAST_DEFAULT_REPORT_SERVER to set a default e.Reporting Server for ReportCast.

How to set the default e.Reporting Server on Windows servers

Use Regedit to create a new registry key on Windows NT and Windows 2000 systems. Before you begin, it is essential to back up your system.

1. From the Start menu, run Regedit.

   Registry Editor appears.

2. Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

3. Choose Edit➛New➛String Value.

   A new value item displays in the right-hand pane of the registry.

4. Type AC_REPORTCAST_DEFAULT_REPORT_SERVER for the name of the new item.

5 Set the default e.Reporting Server:

   1 Select AC_REPORTCAST_DEFAULT_REPORT_SERVER.

   2 Right-click AC_REPORTCAST_DEFAULT_REPORT_SERVER.

   3 Choose Modify.

   Edit String appears.

   4 Type the value of the string in the Value data field.

   5 Choose OK.

For more information about specifying e.Reporting Servers in ReportCast directives, see “Formatting directives” in Chapter 4, “ReportCast directives.”

How to set the default e.Reporting Server on UNIX

1 Use a text editor to edit the startreportcast.sh script.

2 In the script, above the line that calls reportcastsrvr, add lines with the following syntax:

```
export AC_REPORTCAST_DEFAULT_REPORT_SERVER

AC_REPORTCAST_DEFAULT_REPORT_SERVER=<ReportServerName>
```

<ReportServerName> is the name of the default e.Reporting Server.

For example, to set ReportCast's default e.Reporting Server to an e.Reporting Server called Corporate, the lines you add read as follows:

```
export AC_REPORTCAST_DEFAULT_REPORT_SERVER

AC_REPORTCAST_DEFAULT_REPORT_SERVER=Corporate
```

For more information about specifying e.Reporting Servers in ReportCast directives, see “Formatting directives” in Chapter 4, “ReportCast directives.”

Changing the Windows servers ReportCast TCP/IP port

You can change the TCP/IP port that Windows NT and Windows 2000 ReportCast Agent and ReportCast server use to communicate. You must make the change on both the ReportCast server and ReportCast Agent machines. Use regedit to modify the registry key AC_REPORTCAST_PORT in HKEY_LOCAL_MACHINE/SOFTWARE/Actuate/Actuate ReportCast Server/5 on the ReportCast Server machine, and HKEY_LOCAL_MACHINE/SOFTWARE/Actuate/Actuate ReportCast Agent/5 on the ReportCast Agent machine. Before you begin, back up your system.

On both the ReportCast Agent and ReportCast server machines, perform the following steps to change the TCP/IP port:

1. From the Start menu, run Regedit.

   Registry Editor appears.

2. Edit the AC_REPORTCAST_PORT registry keys:

   1. On the ReportCast server machine, choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

      On the ReportCast Agent machine, choose HKEY_LOCAL_MACHINE\SOFTWARE\Actuate\Actuate ReportCast Agent\5.

   2. Select AC_REPORTCAST_PORT.

   3. Choose Edit➛New➛String Value.

      A new value item displays in the right-hand pane of the registry.

   4. Type the new port number.

3. Verify that the value for AC_REPORTCAST_PORT on the ReportCast Agent machine matches the value of AC_REPORTCAST_PORT on the ReportCast server machine.

4. Choose OK.

When you are finished, restart the web server on the ReportCast Agent machine and restart the ReportCast server on the ReportCast server machine.

Changing the ReportCast server machine

You can change the ReportCast server with which a ReportCast Agent communicates. You make the change on the ReportCast Agent machine. Use regedit to modify the registry key AC_REPORTCAST_HOST in HKEY_LOCAL_MACHINE\SOFTWARE\Actuate\Actuate ReportCast Agent\5. Before you begin, back up your system.

Perform the following steps to change the ReportCast server with which a ReportCast Agent communicates:

1. From the Start menu, run Regedit.

   Registry Editor appears.

2. Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Agent\5.

3. Select AC_REPORTCAST_HOST.

4 Choose Edit➛New➛String Value.

A new value item displays in the right-hand pane of the registry.

5 Type the name of the machine that hosts the ReportCast server with which to communicate.

6 Choose OK.

When you are finished, restart the web server on the ReportCast Agent machine.

Using non-default web server ports

The default port for most web servers is 80, or 8080. If your web server uses a non-default port, you must either specify the port as part of the URL, or set up an alias to redirect web browsers to another URL. The ReportCast installation process automatically installs ReportCast to communicate with the web server's port. When you set up links to your reporting web site, however, or define base URLs for accessing files on your reporting web site, specify the web server port.

For example:

```
http://mustique:81/acweb/Sales
```

indicates that the web server runs on a machine called Mustique, using port 81.

See your web server documentation for information about setting up web server aliases and redirecting web browsers.

Configuring ReportCast port numbers

You can specify a range of port numbers for ReportCast to use to connect to the e.Reporting Server. You might need to configure the port numbers if your web server uses a specific port number, such as a firewall.

On Windows NT and Windows 2000, you configure the start of the range of port numbers by setting a registry string value with the name AC_SERVER_SOCKET_BASE. The range of port numbers is set using a string value with the name AC_SERVER_SOCKET_COUNT. The string value names are added to the key HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Report Server\5. The values must be positive integers and the maximum value for both is 32767.

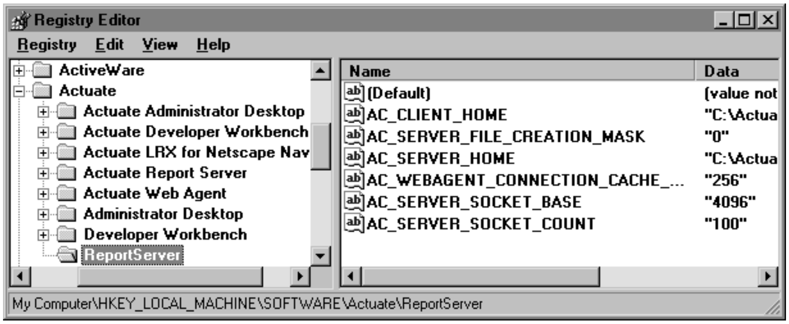
On UNIX, you set the AC_SERVER_SOCKET_BASE and AC_SERVER_SOCKET_COUNT environment variables.

If the start of the range of port numbers is not set, or the start of the range of port numbers is less than 1, the default value is 4096. If the count is not set, the count default value is 8192. If the count is set to less than 1, the count value defaults to 1.

How to configure ReportCast port range on Windows servers

1. If necessary, stop the e.Reporting Server, web server, and ReportCast Server processes. (ReportCast Agent is part of the web server processes.)
2. Start the registry editor REGEDIT.EXE.
3. Navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Report Server\5.
4. Create the string value name AC_SERVER_SOCKET_BASE and the value of first port number.
5. Create the string value name AC_SERVER_SOCKET_COUNT and the value of the range of port numbers.

   This example shows the base port number set to 4096 and the range of port numbers set to 100.



Name	Data
(Default)	(value not
AC_CLIENT_HOME	"C:\Actua
AC_SERVER_FILE_CREATION_MASK	"0"
AC_SERVER_HOME	"C:\Actua
AC_WEBAGENT_CONNECTION_CACHE_...	"256"
AC_SERVER_SOCKET_BASE	"4096"
AC_SERVER_SOCKET_COUNT	"100"

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Actuate\ReportServer

6. Close the registry editor.
7. Restart the e.Reporting Server and web server processes.

For more information about how to set registry keys, see “How to set a new locale registry key on a Windows server system,” later in this chapter.

How to configure ReportCast port range on UNIX

1. If necessary, stop the e.Reporting Server and web server processes. ReportCast is part of the web server processes.
2. Add lines to set the environment variables to e.Reporting Server request server startup script $AC_SERVER_HOME/bin/reqsrvr.sh.

   Add the new line before the first line that uses the export command.

For example, add these lines to change the AC_SERVER_SOCKET base port number to 4096 and the range to 100:

```
AC_SERVER_SOCKET_BASE=4096
export AC_SERVER_SOCKET_BASE
AC_SERVER_SOCKET_COUNT=100
export AC_SERVER_SOCKET_COUNT
```

3 Start the e.Reporting Server.

For more information about how to set environment variables, see “How to set locale environment variable on a UNIX system,” later in this chapter.

Configuring PDF viewing and saving options

Use the AC_REPORTCAST_SEND_INLINE_PDF registry key on NT or environment variable on UNIX to set your users’ PDF options. Setting AC_REPORTCAST_SEND_INLINE_PDF sets the PDF options for all users accessing a particular ReportCast Server. Once you set AC_REPORTCAST_SEND_INLINE_PDF, ReportCast ignores any PDF options submitted through a web browser.

The values for AC_REPORTCAST_SEND_INLINE_PDF are:

- True—View the report in PDF format directly, using the Adobe Acrobat Reader.
- False—Save the report in PDF format to the user’s local disk for printing at another time.

The values true and false are case-insensitive.

ReportCast installation does not set AC_REPORTCAST_SEND_INLINE_PDF. Set AC_REPORTCAST_SEND_INLINE_PDF on the ReportCast Server machine.

To set AC_REPORTCAST_SEND_INLINE_PDF on NT systems:

1 Start the registry editor REGEDIT.EXE.

2 Navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Report Server\5.

3 Create the string value name AC_REPORTCAST_SEND_INLINE_PDF and the value of the PDF viewing or printing option.

4 Close the registry editor.

To set AC_REPORTCAST_SEND_INLINE_PDF on UNIX systems:

1 Use a text editor to edit the startreportcast.sh script.

2. In the script, above the line that calls reportcastsrvr, add lines with the following syntax:

   ```
   export AC_REPORTCAST_SEND_INLINE_PDF
   AC_REPORTCAST_SEND_INLINE_PDF = <PDFOption>
   ```

   <PDFOption> is true to view reports in PDF format, or false to save reports in PDF format to disk.

Creating custom configurations

ReportCast comes with standard configurations such as enu (U.S. English) and jpn (Japanese). The default configuration is called default. To work with custom configurations, you must add <config> to your URL. If you do not add <config>, the ReportCast uses default as the configuration.

The URL has the following syntax:

```
http://<webserver>/acweb/<config>/<reportserver>
```

- <webserver> is your web server name.
- acweb is a literal string indicating a directive for ReportCast.
- <config> is the name of the configuration to use.
- <reportserver> is the name of the e.Reporting Server to access.

How to create a custom configuration

You can create custom configurations, for example, for a particular division or department:

1. Go to the web server's Actuate directory.
2. Add a new directory named for the custom configuration.
3. In the custom configuration directory, add a directory named Standard.
4. Copy the standard ReportCast templates to the custom configuration's Standard directory.

How to set up directories for the custom templates in a custom configuration

1. In the custom configuration directory, create a directory called Custom.
2. Copy the templates you want to customize into the Custom directory.
3. Customize the templates as appropriate.

For more information about customizing templates, see Chapter 5, "ReportCast scripting language reference."

Mapping custom configurations to locales

ReportCast maps the name of its standard configurations to localization settings, also known as locales, to determine how to print numbers and dates and how to sort strings. The differences are significant to users in those language groups. For example, the French, English, and Japanese locales display date and time values differently.

The names of configurations and locales are usually the same.

Locales are standard operating system features. They have standard three-letter names, such as enu, jpn, and fra, for U.S. English, Japanese, and French. When you create a custom configuration that does not have the same name as its locale, you map the custom configuration to the appropriate locale to enable the appropriate functionality. On a Windows NT or Windows 2000 system, you create a new registry key on the machine which hosts the ReportCast Server. On a UNIX system, you set an environment variable on the machine which hosts the ReportCast Server.

Folder items and other lists sorted by string values display in order according to the locale of the machine running ReportCast, not necessarily according to the locale specified by the configuration in the URL.

Setting locales

This section explains how to set locales on Windows NT, Windows 2000, and UNIX systems.

How to set a new locale registry key on a Windows server system

Use Regedit32 to create a new registry key on the Windows NT or Windows 2000 system that hosts the ReportCast Server. Before you begin, it is essential to back up your system.

1 From the Start menu, run Regedit.

   Registry Editor appears.

2 Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

3 Choose Edit➛Add Value.

   Add Value appears.

   1 Type: AC_REPORTCAST_<CONFIG>_LOCALE for the name of the new item.

      <CONFIG>, the name of the custom configuration, must be uppercase.

2 Set the datatype as REG_SZ.

3 Choose OK.

String Editor appears.

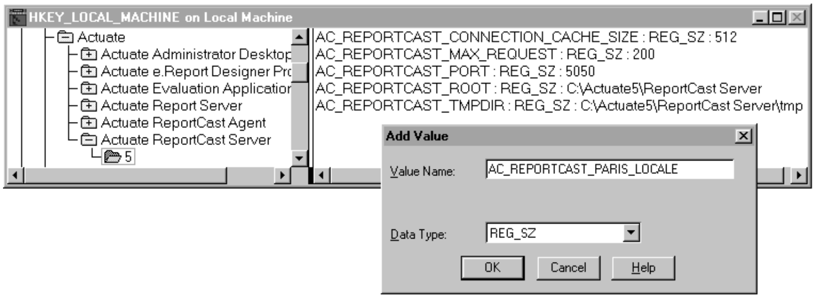
4 Type the locale in the String field.

For example, if the locale is French, type fra. If the locale is U.S. English, type enu.

4 Click OK.

5 Close Regedit32.

For example, if you have a custom configuration for your company's French office called PARIS, and you want the configuration to use the fra locale information, set AC_REPORTCAST_PARIS_LOCALE to fra. The following illustration shows you how to modify the registry key.

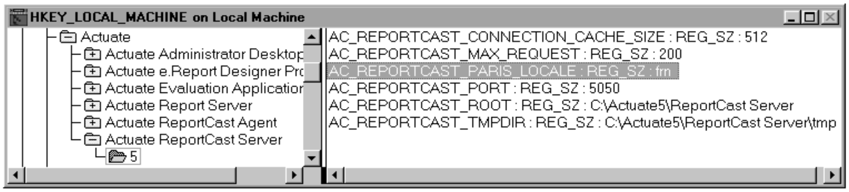

HKEY_LOCAL_MACHINE on Local Machine
Actuate
Actuate Administrator Desktop
Actuate e.Report Designer Pro
Actuate Evaluation Application
Actuate Report Server
Actuate ReportCast Agent
Actuate ReportCast Server
5
Add Value
OK
Cancel
Help

```
AC_REPORTCAST_CONNECTION_CACHE_SIZE : REG_SZ : 512
AC_REPORTCAST_MAX_REQUEST : REG_SZ : 200
AC_REPORTCAST_PORT : REG_SZ : 5050
AC_REPORTCAST_ROOT : REG_SZ : C:\Actuate5\ReportCast Server
AC_REPORTCAST_TMPDIR : REG_SZ : C:\Actuate5\ReportCast Server\tmp
```

Value Name: AC_REPORTCAST_PARIS_LOCALE

Data Type: REG_SZ

The following illustration shows the modified registry key.


HKEY_LOCAL_MACHINE on Local Machine
Actuate
Actuate Administrator Desktop
Actuate e.Report Designer Pro
Actuate Evaluation Application
Actuate Report Server
Actuate ReportCast Agent
Actuate ReportCast Server
5

```
AC_REPORTCAST_CONNECTION_CACHE_SIZE : REG_SZ : 512
AC_REPORTCAST_MAX_REQUEST : REG_SZ : 200
AC_REPORTCAST_PARIS_LOCALE : REG_SZ : frn
AC_REPORTCAST_PORT : REG_SZ : 5050
AC_REPORTCAST_ROOT : REG_SZ : C:\Actuate5\ReportCast Server
AC_REPORTCAST_TMPDIR : REG_SZ : C:\Actuate5\ReportCast Server\tmp
```

How to set locale environment variable on a UNIX system

1 Use a text editor to edit the startreportcast.sh script.

2 In the script, above the line that calls reportcastsrvr, add lines with the following syntax:

```
export AC_REPORTCAST_<config>_LOCALE
AC_REPORTCAST_<config>_LOCALE=<LocaleName>
```

where <config> is the name of the custom configuration

and <LocaleName> is the name of the locale.

For example, to set the BUREAUFRANCE locale on your UNIX system, the lines you add read as follows:

```
export AC_REPORTCAST_BUREAUFRANCE_LOCALE
AC_REPORTCAST_BUREAUFRANCE_LOCALE=fra
```

Supporting locales with e.Reporting Server and ReportCast

The Actuate e.Reporting Server uses a localemap file to support different locales when converting reports in DHTML, XML, or PDF formats. For more information about localemap file, see *Actuate e.Reporting Server Guide*.

ReportCast also uses a localemap file to support different locales. The localemap file is located in the web server's document root directory, for example, <webroot>\docs\actuate\etc.

Any modifications made to the localemap file on an e.Reporting Server system must also be made on any ReportCast system that connects to the e.Reporting Server. The same system locales on the e.Reporting Server system must also be installed on the ReportCast system to allow Actuate locale support to work with Actuate reports in DHTML, XML, and PDF formats.

Accessing UNIX ReportCast with CGI scripts

If your UNIX system uses CGI scripts to access ReportCast, consider the following issues:

- If the ReportCast service is not running and a user attempts to connect to ReportCast, the CGI implementation returns an error. The NSAPI implementation returns a blank page.
- If your web server automatically generates response headers, such as cookies, consider using a parsed-header CGI script to access ReportCast.
- CGI scripts do not automatically resolve escape sequences into the special characters they represent. Your CGI script must include code that translates escape sequences. An escape sequence begins with the percent character (%), followed by the ASCII two-digit hexadecimal code for the special character. For example, add code to translate the following URL:

  ```
  http://RS%22Main/acweb/SalesRpts
  ```

  to:

  ```
  http://RS_Main/acweb/SalesRpts
  ```

CGI scripts are supported only with UNIX ReportCast. There is no CGI counterpart supported for Windows NT and Windows 2000.

How to use a parsed-header CGI script

1. Export the environment variable:

   HTTP_AC_REPORTCAST_PARSED_HEADERS

2. In the CGI script, set the environment variable to True.

   The ReportCast installation process generates two CGI scripts:

   - nph-actuate.cgi
   - ph-actuate.cgi

   The script ph-actuate.cgi is a parsed-header script. A URL that contains this script name generates output to which the web server can add its own headers.

If you use a parsed-header CGI script to access ReportCast, your URL must reference ph-actuate.cgi to produce parsed-header output.

Working with reports on Windows web sites

To work with reports on a Windows-based web reporting site, register a browser plug-in or helper application for the MIME types of Report Encyclopedia items. You can register plug-ins and helper applications in your browser. For example, in the Netscape browser, register a helper application from Options➛General Preferences➛Helpers.

If no browser plug-in or helper application is registered for a requested Report Encyclopedia item, the web browser prompts you to save the item to the local file system. The default file name provided in the Save dialog box includes a version number, for example, Report.roi;1. Windows systems use the filename extension to recognize the file type. To ensure that Windows can recognize the file type, delete the version part of the filename, including the semicolon. In this example, you save the file as Report.roi because Windows does not accept the ;1 convention. You can also modify the name to something like Report.roi to add versioning information.

If the name of the Report Encyclopedia item contains spaces or other special characters, the default file name contains character sequences beginning with % instead of the actual characters.

Using CGI scripts for security

You may need to mask the location on the report server of reports that a user views on the web reporting site.

For example, a custom HTML page can display a link to a report at:

```
http://webserver/cgi/secure.cgi/myreport
```

instead of the actual location. For example:

```
http://webserver/cgi/secure.cgi/acweb/reportserver/accounts/trusted/
myreport.roi?View
```

The CGI script masks the report URL on the web reporting site. To mask the report URLs and use the CGI script to reconstruct the complete URLs:

- Set the CGI script to change the incoming, disguised values of the PATH_INFO and QUERY_STRING CGI variables to construct a valid report-viewing request URL.
- Set the CGI script to set and export the environment variable HTTP_AC_REPORTCAST_BASE_URL to the unmasked URL.

  The Base URL is that part of the URL up to and including the name of the machine running the report server through which the report is normally accessed on the web reporting site.

In this example:

- The incoming value of PATH_INFO is /myreport. The CGI script resets it to /acweb/reportserver/accounts/trusted/myreport.roi.
- The incoming value of QUERY_STRING is empty. The CGI script resets it to View.
- The Base URL value is /cgi/secure.cgi/acweb/reportserver, or, alternatively, http://webserver/cgi/secure.cgi/acweb/reportserver.

Administering ReportCast channels

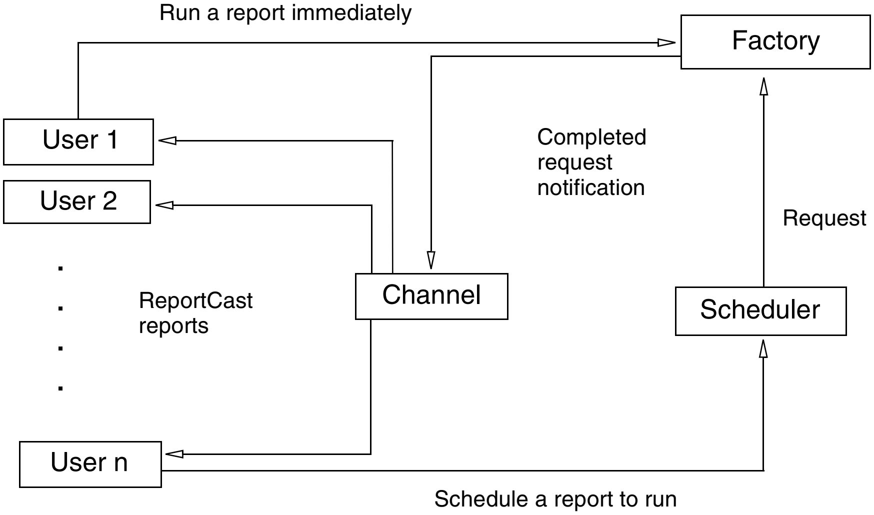
This section describes how to administer ReportCast channels.

About Actuate ReportCast channels

Actuate ReportCast channels is a service that allows users to subscribe to particular web channels and view reports available on those channels. ReportCast channels uses push technology to deliver reports to particular web channels. Push technology is also known as web casting, push distribution, or publish and subscribe. When new reports become available, the appropriate channel receives a notification. Subscribed users check the channel's web page to get a listing of available reports. When users click on a report link, the report appears in the web browser window.

ReportCast channels allows you to subscribe to and unsubscribe from particular channels, list the channels to which you are currently subscribed, and view the contents of those channels. If you have administrator privileges, you can also create, delete, and modify channels.

The following illustration shows how ReportCast channels fit into the Actuate Internet architecture.



Users make requests to run reports immediately or at a scheduled time. The requester form allows users to select the users and channels to notify when the report is complete. Once the report runs successfully, the completed request notification goes to the appropriate users and channels.

For more information about factory processing, see Chapter 8, "Understanding the report generation process," in *Programming e.Reports*. For more information about scheduling reports, see Chapter 4, "Running a report from the desktop," in *Using e.Reports*.

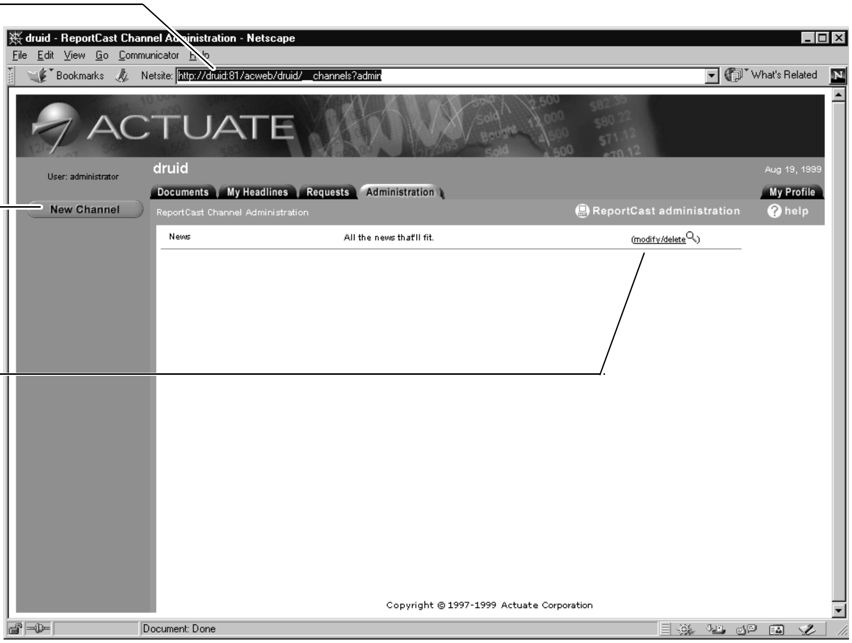
Users with administrative privileges on e.Reporting Server can administer ReportCast channels. Choose the Administration tab on the Report Encyclopedia web page, or use the _ _channels?admin directive to access the channel administration page. The complete syntax for the directive is:

```
http://<webserver>/acweb/<reportserver>/_ _channels?admin
```

The following table lists the ReportCast channels administrative directives. All commands are available from the channels administration page.

Command	Description
_ _channels?admin	Accesses the ReportCast channels administration web page.
_ _completed	Lists the channels to which this user subscribes.
_ _channels/<channel>	Lists the contents of channel <channel>.
_ _channels?retrieveForm	Retrieves and displays the channels subscription form.
_ _channels?submitForm	Submits the subscription form to ReportCast.
_ _channels/<channel>?drop	Removes the specified channel from the e.Reporting Server.
_ _channels/ <channel>?retrieveDetails	Displays the details form for the specified channel. Users with administrator privileges can modify the channel using this form.
_ _channels/ <channel>?submitDetails	Submits the form that modifies details of a channel.
_ _channels?retrieveCreation	Displays the form that allows users with administrator privileges to create a new channel.
_ _channels?submitCreation	Submits the form that creates a new channel.

The following illustration shows the channels administration page.


druid - ReportCast Channel Administration - Netscape
File Edit View Go Communicator Help
Bookmarks
Netsite: http://druid:81/acweb/druid/__channels?admin
What's Related
ACTUATE
User: administrator
druid
Aug 19, 1999
Documents
My Headlines
Requests
Administration
My Profile
New Channel
ReportCast Channel Administration
ReportCast administration
help
News
All the news that'll fit.
(modify/delete)
Copyright © 1997-1999 Actuate Corporation
Document: Done

Directive that generates the channels admin page.

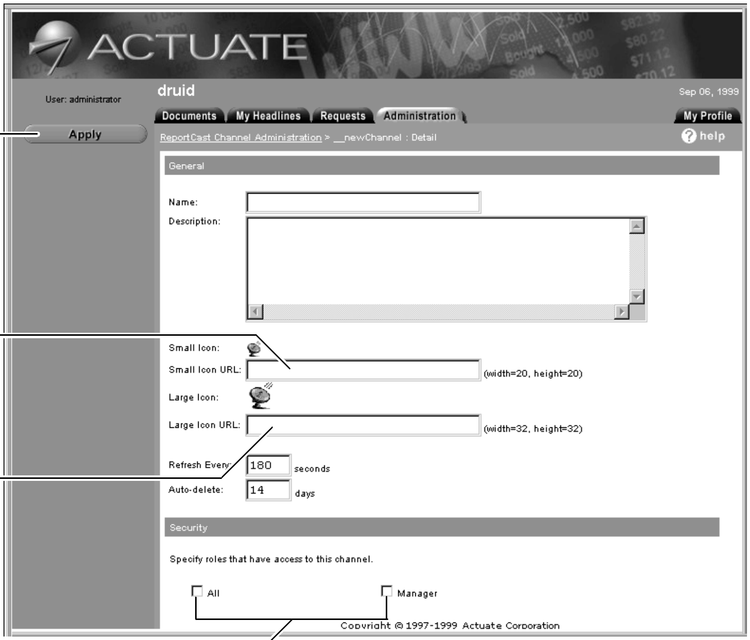
Choosing the New Channel button generates the retrieveCreationdirective.

Click Modify/Delete to make changes to or to remove the channel.

Creating a new channel

This section explains how to create a new ReportCast channel.

When you click the New Channel button, the Channel Administration page appears, as shown in the following illustration.



Refresh Every: 180 seconds

Auto-delete: 14 days

Click Apply to create the new channel.

Enter the URL for the scaled-down image to use for this channel's icon.

Enter the URL for the large image to use for this channel's icon.

Select a Security checkbox to set the privileges needed to access the new channel.

How to create a new channel

To create a new channel from the channels administration page:

1 Choose New Channel.

The new channel creation page displays in the browser.

2 Fill out the channel creation form:

   1 Name is the name of the channel as it appears in channel subscription lists.

   2 Description is text describing the channel's purpose, audience, or other pertinent information.

   3 Small Icon URL is the URL to an image, such as a GIF or JPG, to be used as the small-scale icon for this channel. The default is the satellite dish icon.

   4 Large Icon URL is the URL to an image, such as a GIF or JPG, to be used as the large icon for this channel. The default is the satellite dish icon.

5. Refresh Every determines how often the channel's contents should be updated.
6. Auto-delete determines how long an item may remain in the channel before it is automatically deleted.
7. Use the security section to determine which roles may access the channel.

3. Click Apply to create the channel.

   A confirmation page appears in the browser, showing the new channel's name and details.

Modifying an existing channel

To modify an existing channel from the channel's administration page:

1. Click Modify/Delete for the channel you wish to modify.

   The channel's Details page displays in the browser.

2. Edit the channel's information on Details.
3. Click Apply.

   The updated channel's Details page appears in the browser.

Deleting an existing channel

To delete an existing channel from the channel's administration page:

1. Click Modify/Delete for the channel you wish to modify.

   The channel's Details page appears in the browser.

2. Click Delete Channel.

   The Successful command completion page appears in the browser.

URLs in e-mail notification

Report Encyclopedia administrators can configure the e.Reporting Server to send an e-mail notification when a request is complete. Using ROTP syntax to specify the report's location is the default. ROTP syntax is a report object protocol that runs a report on a server. For example, an e-mail sent to users that contains the line:

```
rotp://bimini/Sales/goals/quarterly.roi;7
```

specifies the seventh version of the report Quarterly.roi is in the folder Sales\Goals in the Report Encyclopedia on Bimini.

As administrator, you can configure the e.Reporting Server to specify a URL for the location of the report when notifying users by e-mail. For example, if users can access the Report Encyclopedia using ReportCast on the web server Paradise, you can configure the e.Reporting Server to specify:

```
http://paradise/acweb/bimini/Sales/goals/quarterly.roi;7?view
```

For more information about specifying URLs, see Chapter 6, "Administering the Report Encyclopedia in the web environment," in *Administering the Report Encyclopedia*.

Tuning ReportCast performance

This section describes some considerations and options for tuning ReportCast performance.

Setting user connection timeouts

Each time a user logs on to the web reporting site, ReportCast creates a connection between itself and the e.Reporting Server. The connection stays open indefinitely unless you specify a timeout period.

Registry keys on Windows NT and Windows 2000, or environment variables on UNIX, allow you to set the timeout period:

- AC_REPORTCAST_CONNECTION_CACHE_SIZE sets the maximum number of e.Reporting Server connections that ReportCast holds open at any time. Set AC_REPORTCAST_CONNECTION_CACHE_SIZE to the number of actual users that will be accessing ReportCast, not necessarily the number of distinct e.Reporting Server users. If the connection cache size is not set, the number of connections is unlimited. Set AC_REPORTCAST_CONNECTION_CACHE_SIZE to 0 (zero) to ensure that connections are never cached from one request to another. The default value is 256.
- AC_REPORTCAST_CONNECTION_CACHE_TIMEOUT sets the approximate time, in seconds, after which an idle e.Reporting Server connection is dropped. If connection cache timeout is not set, connections remain on the e.Reporting Server indefinitely.

On Windows NT and Windows 2000, set the registry keys as string values in HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5. Use Regedit to set the registry keys. For more information about how to set registry keys, see "How to set a new locale registry key on a Windows server system," and "How to set locale environment variable on a UNIX system," earlier in this chapter.

Setting request timeouts

You can set the length of time that ReportCast waits before a request times out. The AC_REPORTCAST_REQUEST_TIMEOUT registry key on Windows NT and Windows 2000 or environment variable on UNIX specifies the request timeout period in seconds. It affects only those report requests for which users checked the Wait checkbox (wait for completion). If AC_REPORTCAST_REQUEST_TIMEOUT is set and the Wait checkbox is checked on the requester form, ReportCast waits the specified number of seconds. If the report request has not completed, ReportCast displays the request status page, rather than waiting for the request to complete.

The default is that there is no timeout.

On Windows NT and Windows 2000, set the registry keys as string values in HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5. Use Regedit to set the registry keys. For more information about how to set registry keys, see "How to set a new locale registry key on a Windows server system," and "How to set locale environment variable on a UNIX system," earlier in this chapter.

Tuning viewing performance

If your users are waiting too long for reports to display over the net, set the DemandCachePageSize registry key to tune performance.

DemandCachePageSize controls the size of data packets sent to an Actuate Viewer LRX over the network. Specify the data packet size in bytes. Packet size must be a multiple of 2048:

- Minimum packet size is 2048 bytes
- Maximum packet size is 128K bytes
- Default packet size is 64K

Consider the following items when deciding how to set the DemandCachePageSize registry key:

- Smaller packet sizes cause more reads to be made to the network. If you have a fast network, larger packet sizes may work well for your users.
- Larger packet sizes cause fewer reads to be made to the network. In general, read time is shorter than connect time.

You may need to experiment to find the packet size that works best for your users. Variables such as network speed, number of users, and so on affect how much particular packet sizes increase report viewing efficiency for your users.

How to set the DemandCachePageSize registry key

For more information about using Regedit, see “How to set a new locale registry key on a Windows server system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

1 Run Regedit:

- For the Netscape browser, navigate to HKEY_CURRENT_USER➛Software➛Actuate➛ Netscape LRX 5.0➛General
- For the Microsoft Internet Explorer browser, navigate to HKEY_CURRENT_USER➛Software➛Actuate➛ MSIE LRX 5.0➛General

2 Select DemandCachePageSize.

3 Set the value of DemandCachePageSize.

Viewing files using the native web server file system

Some web server-based viewing technologies expect to be able to directly access a file by reading it from the web server’s file system. You can store non-Actuate objects in the Report Encyclopedia and view them on the Web using their intended viewing technology.

1 In the Administrator Desktop, enable the Use Web Server File System feature. For information about enabling the Use Web Server File System feature, see *Administering the Report Encyclopedia.*

2 Ensure that the viewing technology is installed on the machine hosting the ReportCast Agent.

3 Use your web browser to access the e.Reporting web site.

4 View a report of the file type for which you enabled the Use Web Server File System feature. Users notice only that the report appears in its native viewer, rather than in Actuate’s viewer or DHTML formats.

ReportCast copies the report to a temporary directory on the web server’s file system for viewing. The temporary directory, ViewTemp, resides in the Actuate directory in the web server’s document root directory. ReportCast disables automatic indexing for the ViewTemp directory for the default installation.

ReportCast generates random, unique file names for the reports. The names have no meaning in relation to the reports. For example, V1_f49_f99808e4_sgth8ft3_1c0-93-5.rpt.

Configuring viewing options

ReportCast administrators configure viewing options by setting environment variables on UNIX systems, or registry keys on Windows NT and Windows 2000 systems.

The following table lists and describes the configuration options for viewing reports on the web server's native file system.

Native web server file system viewing option	Description
AC_REPORTCAST_ VIEWTMPDIR	The full pathname of the temporary directory. The directory must exist in the Actuate directory of the web server document root directory structure.
AC_REPORTCAST_ VIEWTMPDIR_VIRTUAL	The location of the temporary directory relative to the document root of the web server. ReportCast uses this value to hide the actual location of the temporary directory from users. AC_REPORTCAST_VIEWTMPDIR_VIRTUAL indicates how to get to the directory from a web browser. This path can be completely different from the physical location. ReportCast Agent uses this value to redirect the browser for viewing the report.
AC_REPORTCAST_ VIEWTMPDIR_MAX_SIZE	The maximum allowed size of the temporary directory, in kilobytes. When this value is not set, there is no limit.
AC_REPORTCAST_ VIEWTMPDIR_MAX_FILE	The maximum number of files allowed in the temporary directory. When this value is not set, there is no limit.
AC_REPORTCAST_ VIEWTMPDIR_FILE_ PERSISTENCY	The length of time the file remains in the temporary directory, in seconds. The default is 900, or 15 minutes. The minimum value is 60.
AC_REPORTCAST_ VIEWTMPDIR_ CLEANING_FREQUENCY	The frequency with which the automatic cleaning process removes files from the temporary directory, in seconds. The default value is 30.

Files remain in the temporary directory between the value of AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY and the value AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY + AC_REPORTCAST_VIEWTMPDIR_CLEANING_FREQUENCY. The value of AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY must be higher than AC_REPORTCAST_VIEWTMPDIR_CLEANING_FREQUENCY.

Handling viewing security issues

Report Encyclopedia security does not apply while the report is on the web server's hard drive. To help prevent unauthorized access of reports, take the following precautions:

- Restrict access to the temporary directory or restrict logon to the web server host machine.
- Ensure that the directory listing is disabled to prevent users who know the name of the directory from accessing it over the Web.
- Users guessing the temporary file names is an unlikely scenario because the temporary file names are not obvious.

Chapter

3

Working with ReportCast templates

This chapter contains the following topics:

- About ReportCast templates
- Creating custom templates

About ReportCast templates

As you submit directives, ReportCast generates reporting web pages dynamically from HTML templates. The template files contain scripting language commands embedded in HTML comments. The scripting language commands tell ReportCast how to generate and format its web pages.

You can customize the templates to meet your needs. Only the downloading and viewing directives cannot use custom templates. You can customize default forms, perhaps modifying a standard form to display only some of the available parameters, to add a background color or graphic, or to display your company logo.

Actuate provides a set of default, standard templates with the ReportCast Server installation.

To customize ReportCast templates, you should have knowledge of the following:

- Directives you can send to ReportCast. For information about ReportCast directives, see Chapter 4, "ReportCast directives."
- Contents of the template files. For information about template files, see "About HTML template files," later in this chapter.
- ReportCast scripting language. For information about ReportCast scripting language, see Chapter 5, "ReportCast scripting language reference."
- HTML, the HyperText Markup Language. Information about HTML is available from various sites on the World Wide Web and in many books available at bookstores.
- JavaScript. Information about JavaScript is available from various sites on the World Wide Web and in many books available at bookstores.

About HTML template files

All ReportCast template files must have the extension .achtml, for Actuate template HTML. Template files can be specific to display the contents of a particular folder on a particular report server. Template files can also be general to display e.Reporting Server objects on a particular web server. Whether a template applies to one e.Reporting Server folder or to the entire web server depends on its placement in the e.Reporting Server, the web server, or the ReportCast Server directory structure.

Custom HTML requester forms are the exception. These are HTML files that you create to run specific reports. ReportCast does not provide these forms. Custom HTML requester forms must:

- Reside in the same e.Reporting Server folder as the report object executable (.rox) file

- Have the file extension .html
- Depend upon the ROX file

Custom HTML requester forms that reside on the e.Reporting Server in a Report Encyclopedia folder are static. Because custom HTML requester forms reside in the Report Encyclopedia, ReportCast cannot fill in variables such as the e.Reporting Server name and the current date and time.

For more information about the placement of template files, see “About template placement and scope,” later in this chapter.

For more information about creating custom HTML requester forms for specific reports, see “Creating custom requester forms for a specific report,” later in this chapter.

The following table lists and describes the web pages that ReportCast generates.

Generated HTML page	Template file name	Description
Folder Listing	Folderlist.achtml	Contents of an Encyclopedia folder.
Requester Menu	Requestermenu.achtml	Request form that runs reports.
Standard Requester	Request.achtml	Standard request form.
Custom Folder	Index.achtml	Custom folder page created by an administrator and stored directly on an e.Reporting Server.
Request Folder	Requestlist.achtml	Contents of one or all request folders.
Request Confirmation	Confirm.achtml	Response page returned after a user submits a request to run a report.
Request Detail	Requestdetail.achtml	Response page used to display detailed information about a request.
Request Status	Status.achtml	Status page returned after a user asks for request status.
Canceled Request Confirmation	Cancel.achtml	Response page returned when a request to cancel an active request is successful.

Generated HTML page	Template file name	Description
File Detail	Filedetail.achtml	Lists file properties.
Confirmation	Success.achtml	Response page returned when a request to drop or rename a report is successful.
Error	Error.achtml	Response page generated after an error condition is detected.
Administration	Admin.achtml	Page generated to perform general, non-e.Reporting Server-related, administrative tasks.
Channel Subscription	Channelsubscription. achtml	Subscription form to subscribe to and unsubscribe from channels.
Channel Administration	Channeladmin.achtml	Top-level channel administration page.
Channel Contents	Channelcontents.achtml	Page that displays contents of a specified channel.
Channel Detail	Channeldetail.achtml	Administrators use this page to create, delete, and modify a specified channel.
Report Viewing Options	Userpreferences.achtml	Choose how to view reports.
View Frameset	Viewframeset.achtml	The initial set of frames that display the DHTML report, including the navigation toolbar, search window, table of contents, and the DHTML report.
Navigation	Viewnav.achtml	The navigation toolbar used for viewing the DHTML report.
Table of Contents	Viewtoc.achtml	The DHTML report's table of contents.
File Versions	Fileversions.achtml	Display the list of available versions of the current file.

About internal template structure

The standard ReportCast pages include many elements that several templates use, such as the copyright section and page headers. The ReportCast templates directory contains files with names such as Contentactive.achtml, Contentbegin.achtml, and Contentfileversions.achtml. These subtemplate files generate elements common to many ReportCast pages. ReportCast templates include these subtemplate files, using ReportCast scripting language include statements, to provide elements such as the copyright notice, page headers, and folder contents listing. To create custom templates, customize the individual subtemplate files as well as the standard template files.

About template placement and scope

A template's placement determines its scope. A template can apply to a specific report, report server folder, or to an entire web server.

You can place templates in the Report Encyclopedia or on the web server. Templates placed in the Report Encyclopedia do not have access to the ReportCast variables such as the current date and time. Placing templates in the Report Encyclopedia, however, supports customizing specific folders and creating requester forms for specific reports.

Placement in an e.Reporting Server folder for a specific report

Placing an HTML requester form in an e.Reporting Server folder and creating a dependency from that requester form to a particular report object executable (.rox) file provides a custom requester form for that specific report executable. If more than one HTML file in the e.Reporting Server folder depends upon a particular .rox file, ReportCast displays a requester menu form. The user chooses the custom requester to use.

Placement in an e.Reporting Server folder

Placing a template called Index.achtml in an e.Reporting Server folder provides a folder-specific template. Such templates apply only to a particular folder's contents on a particular e.Reporting Server.

Placement in an e.Reporting Server's _ _WebAgent folder

Creating a _ _WebAgent folder on an e.Reporting Server and placing ReportCast templates inside provides e.Reporting Server-specific templates. Such templates apply only to objects on a particular e.Reporting Server.

Placement in the Custom directory

Placing ReportCast templates in a directory called Custom in the ReportCast Server directory tree provides configuration-specific templates. Such templates apply only to objects for a particular ReportCast configuration, such as the Japanese-language or French-language configuration.

Locate the Custom directory in the Actuate distribution's ReportCast Server directory:

```
<Actuate distribution>\Reportcast server\<Config>\Custom
```

where <Config> is the name of the configuration. For example, if your Actuate distribution is on drive C in the Actuate5 directory and you are providing a Japanese-language configuration, the directory path is:

```
C:\Actuate5\Reportcast server\Jpn\Custom
```

If your UNIX ReportCast Server is in the Actuate distribution directory /usr/actuate and you are providing a Japanese-language configuration, the directory path is:

```
/usr/actuate/ReportCast/templates/jpn/custom
```

If you plan to provide custom templates for a particular configuration, you must create the Custom directory yourself and place the custom templates in it.

Placement in the Standard directory

ReportCast templates located in the \Actuate5\Reportcast server\<Config>\Standard directory are the default ReportCast templates supplied by Actuate. For UNIX installations, if the Actuate distribution directory is /usr/actuate, the path to the default templates is:

```
/usr/actuate/ReportCast/templates/<config>/standard
```

where <config> is the name of the configuration.

Search order for web templates

ReportCast searches for template files in a specific order:

- e.Reporting Server folder
- e.Reporting Server's _ _WebAgent directory
- The Custom directory for a particular configuration
- The Standard directory for the configuration

For more detailed information about the search order, see "Understanding ReportCast searches for templates," later in this chapter.

About the ReportCast object model

To understand how ReportCast's directives, scripting language commands, and templates work together, it is useful to understand something about the objects with which ReportCast works. ReportCast objects represent the URL requests, web server, e.Reporting Server, and objects on the e.Reporting Server. ReportCast performs actions on e.Reporting Server objects or displays their properties.

ReportCast objects

ReportCast operates on these types of objects.

Object	Description
Web server	Web server running ReportCast.
ReportCast	ReportCast process.
Web request	Request that ReportCast is currently handling.
e.Reporting Server	Actuate e.Reporting Server on which the requested object is located.
Folder	Folder in a Report Encyclopedia. Folders can contain other folders as well as files.
Folder item	One of the contents of a folder.
Report definition	ROX, ROV, ROI, and ROW file types.
Request folder	A folder containing active, scheduled, or completed requests.
Request	A request to run a report. Requests are located in Request folders.
Member	A user, role, or group.
Parameter group	A group containing parameters for a specific report request.
Parameter	A specific parameter used to create a request.
Schedule	The part of a request's schedule that describes the times at which the request is to run.
Printer property	The part of the print request that provides information about the printer that will print the requested report.
Channel	An object that contains a set of requests.
Error	An object that contains information about an error encountered by ReportCast.

Object	Description
TOC item	An object that consists of a displayable name and a component ID.
View session	An object that tracks the current page, navigation mode, format, and other viewing properties for the current report.

Working with the ReportCast object model

The following table shows the relationships among ReportCast objects. The second column lists the objects with which the objects in the first column are associated.

Object	Provides a list of
Web server	Nothing
ReportCast	Nothing
Web request	Nothing
e.Reporting Server	Members, request folders, channels
Folder	Folder items, members (those granted privileges)
Folder item	Depends on the kind of item
Report definition	Parameters, parameter groups, parameter values
Request	Request status
Request folder	Requests
Member	Privileges within the context of a folder item or channel
Parameter	Nothing
Parameter group	Parameter
Parameter value	Nothing
Privilege	Nothing
Channel	Requests
Error	Nothing
View session	The current page being viewed, the component ID of the current Table of Contents entry, the total number of pages in the DHTML report, and the format for searching or viewing.

Object	Provides a list of
TOC item	The current table of contents (TOC) item’s component ID, an indicator showing whether or not the current TOC item is expandable, and the name of the current TOC item.
Search criterion	The search criterion’s class, the name of the control and the component ID to which the current criterion applies, the search expression, and an indicator showing whether or not to include the selected control with the search results.
Search result set	Total number of hits, number of hits displayed, the index of the first hit and last hits displayed, the format URL, and the frameset URL.
Search result field	The current search result hit’s component ID and the displayable value of the current search field.

Understanding objects available to a template

When you create templates, it is convenient to know when particular objects are available.

These objects are always available:

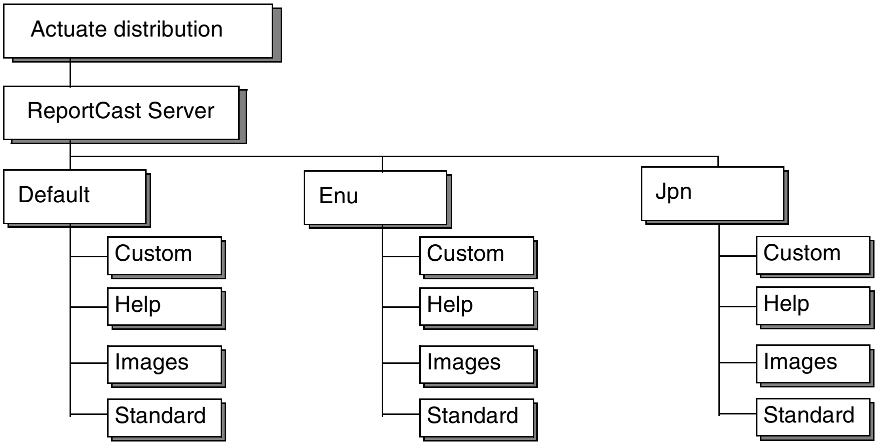
- e.Reporting Server (if specified in the URL)
- ReportCast
- Web request
- Web server

The following rules determine when other objects are accessible. If the object does not meet one of the following requirements, it is not accessible:

- If an object is on the web page of the web server, ReportCast, request, or e.Reporting Server, it is available for that page’s template.
- List objects are available to list all objects related to a specific object, such as a list of privileges, e.Reporting Servers, and so on. Each iteration through the list loop makes the next object in the list available.
- If an object is accessible, you can make its variables and list statements accessible as well using the With command. The With command has no effect if the object is not already accessible. For more information about the With command, see “With command,” in Chapter 5, “ReportCast scripting language reference.”

About the web server template directory structure

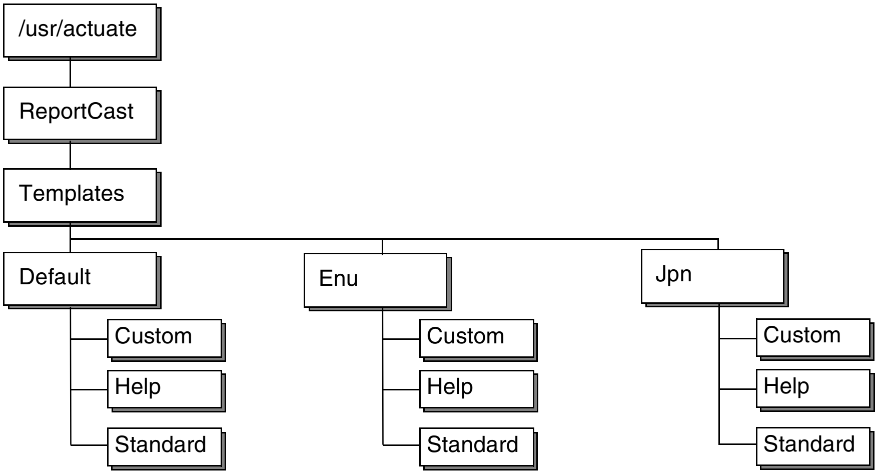
When you install ReportCast, the installation process sets up a template directory structure. The ReportCast Server structure is similar to the following diagram.



In this diagram, Actuate distribution represents the directory structure of your Actuate installation. For example, C:\Actuate5.

The directories jpn and enu are standard abbreviations for localized languages. The directory Default is the default ReportCast configuration. If you do not specify a configuration in the reporting web URL, ReportCast uses the Default directory.

For UNIX ReportCast Server installations, the structure is similar to the following diagram.



In this diagram, /usr/actuate represents the root of your Actuate distribution directory structure.

ReportCast installations have the following additional considerations:

- Static content, such as JavaScript files and images, reside on the web server with the ReportCast Agent.
- The ACHTML template files reside in the Actuate distribution directory structure with the ReportCast Server.
- Customize JavaScript files and images on the ReportCast Agent machine, and customize templates on the ReportCast Server machine.

The installation process automatically creates a directory structure for each language that ReportCast supports. The default template files go in the standard directories. Copy the templates you want to customize to the appropriate custom directory, and make changes to them there. This process ensures that you always have the original default templates. The help directories contain online help files for ReportCast. The images directories contain the images and icons that display on the reporting web pages, such as the folder, schedule, and search icons.

Creating and customizing ReportCast template files

Create and customize ReportCast template files in the same way that you work with any web page. Use an editing tool or an HTML authoring tool. Most authoring tools do not support the ReportCast scripting language. You can use authoring tools to design your ReportCast pages, then add the scripting commands later. If your authoring tool supports adding HTML comments to your web pages, use that feature to add the scripting commands.

You cannot modify the following JavaScript files:

- Report.js
- Converter.js
- Layer.js
- Array.js
- Reportstyle.css

The paths to these JavaScript files' locations are hard-coded in ReportCast.

Template file elements

The default HTML template files contain these elements:

- HTML tags that determine text format
- Text

- Variables, which ReportCast substitutes with data according to the directive you issue. These variables appear in the following format:

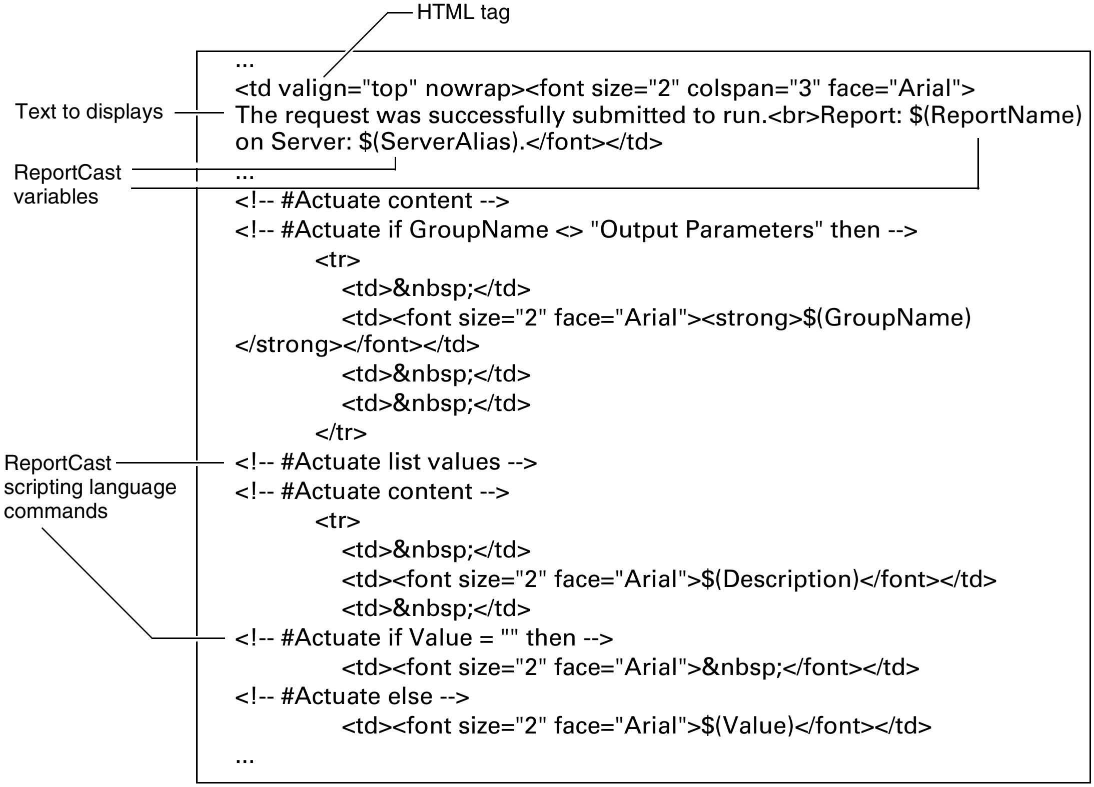
  $ (VARIABLENAME)

- ReportCast scripting language directives. These directives appear in the following format:

  <!-- #Actuate <directive> [options] -->

- JavaScript

The following is a fragment of the HTML document source for the Contentconfirm.achtml file, a template file that generates the content portion of the request confirmation web page. The request confirmation page appears after a user submits a request to run a report.



```
...
<td valign="top" nowrap><font size="2" colspan="3" face="Arial">
The request was successfully submitted to run.<br>Report: $(ReportName)
on Server: $(ServerAlias).</font></td>
...
<!-- #Actuate content -->
<!-- #Actuate if GroupName <> "Output Parameters" then -->
         <tr>
            <td>&nbsp;</td>
            <td><font size="2" face="Arial"><strong>$(GroupName)
</strong></font></td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
         </tr>
<!-- #Actuate list values -->
<!-- #Actuate content -->
         <tr>
            <td>&nbsp;</td>
            <td><font size="2" face="Arial">$(Description)</font></td>
            <td>&nbsp;</td>
<!-- #Actuate if Value = "" then -->
            <td><font size="2" face="Arial">&nbsp;</font></td>
<!-- #Actuate else -->
            <td><font size="2" face="Arial">$(Value)</font></td>
...
```

Modifying HTML template files

You can easily modify template files written in HTML to change the content and format of the HTML pages ReportCast generates. For example, you can change the background pattern or color or replace the Actuate bitmap on the HTML page. You can also add or delete variables in the template files to change the data.

Before editing any of these template files, you should:

- Be well-versed in HTML.
- Understand how to use ReportCast variables in templates.
- Be familiar with the ReportCast scripting language.
- Be familiar with JavaScript.

For more detailed information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.” For more detailed information about descriptions of the ReportCast scripting language, see Chapter 5, “ReportCast scripting language reference.”

Locating ReportCast template files

ReportCast expects templates to:

- Be in specific locations
- Have specific names

If you put your templates in different locations, or give them different names, ReportCast will not find them. If ReportCast does not find the appropriate template, that template cannot be used to generate web pages. If you change the name of an included subtemplate file, you must change all references to that subtemplate file in all the ReportCast templates that refer to it. For more information about naming ReportCast template files, see “Naming template files,” later in this chapter.

Understanding ReportCast searches for templates

When it receives a directive, ReportCast first determines whether to look for custom templates. ReportCast then searches for custom templates in a specific order:

1. Determine the search type. The request can be to display folder contents or to run a report:
   - If this is a request to display folder contents, look in the e.Reporting Server folder for an Index.achtml file. If there is an Index.achtml file, use it to generate the folder contents web page.

- If this is a request to run a report, look in the e.Reporting Server folder for an HTML file that depends on the report's .rox file. If there is such a file, assume it is a custom requester form for the report and display it as the requester web page. If there is more than one HTML file that depends on a particular .rox file, display a requester menu form. The user chooses the form.

2 Search for template files in the e.Reporting Server's _ _WebAgent folder.

Templates in the _ _WebAgent folder generate customized web pages for this e.Reporting Server.

3 For installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server's Custom directory for the configuration currently in use. For installations with ReportCast Agent and ReportCast Server on separate machines, search for template files in the ReportCast Server's \Templates\<Config>\Custom directory in the Actuate distribution directory, where <Config> is the configuration currently in use.

Templates in the Custom directory generate web pages for this particular configuration. For example, if the configuration in use is NT ReportCast's French configuration, ReportCast uses the custom templates in \wwwroot\Actuate\Fra\Custom to generate French-language web pages.

4 For installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server's Standard directory. For installations with ReportCast Agent and ReportCast Server on separate machines, search for template files in the ReportCast Server's \Templates\<Config>\Standard directory in the Actuate distribution directory, where <Config> is the configuration currently in use.

If no other template files exist, ReportCast uses the default templates in the Standard directory.

You can develop custom templates for a particular e.Reporting Server or web server to create a distinctive look for a departmental e.Reporting Server or web server. Custom templates ensure that the department's web pages have a consistent design across your company's intranet.

The default template files are located in the ReportCast server's \Reportcast\ Templates\Default\Standard directory of your Actuate distribution directory structure.

The following table lists the expected locations for ReportCast template files. Docroot refers to the document root, whether on the web server, or in the

ReportCast Server directory structure Actuatedir refers to the Actuate directory structure of your Actuate distribution.

Location	Description
\Actuatedir\Reportcast Server\[Config\]Standard	Location of the default templates for configuration [Config] of this web server. The default directory contains the default Actuate ReportCast configuration.
\Actuatedir\Reportcast Server\[Config\]Custom	Location of custom templates that you create for configuration [Config] of this web server.
_ _WebAgent	Location of custom templates that you create for this e.Reporting Server. Use the e.Reporting Server Navigator to create this folder on your e.Reporting Server.
Index.achtml	The custom template you create specifically for a particular folder on your e.Reporting Server. Use the e.Reporting Server Navigator to place this template file in the appropriate folder on your e.Reporting Server.

A template's location—in an e.Reporting Server's __WebAgent folder, in a custom directory, and so on—determines its scope. For more information about placement of templates for customizing part or all of the ReportCast interface, see "About template placement and scope," earlier in this chapter.

Creating custom requester forms for a specific report

You can create custom requester forms for a particular report. If an .rox or an .rov file has one or more HTML file that depends on it, ReportCast treats the file as a custom requester form. When a request to run one of those reports occurs, ReportCast displays the custom requester form rather than the standard form. If more than one custom requester form exists, ReportCast displays a menu and allows the user to choose the form to use.

How to create a custom requester form

1 Use the editing tool of your choice to create a custom HTML requester form. You can use the standard requester, Request.achtml, as a model.

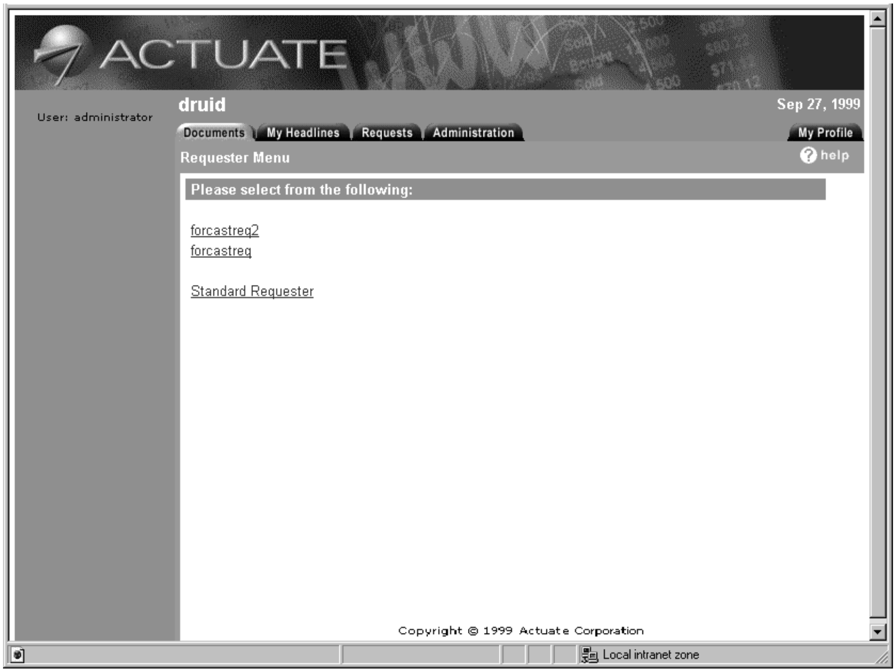
2 Save the requester form with the extension .html.

3 In the Administrator Desktop, navigate to the e.Reporting Server folder containing the report. That is where you attach the custom HTML requester form.

How to attach a custom requester form to a report

1. Right-click the e.Reporting Server folder that contains the report. In the context menu, choose New Report Item.
2. In Save To Volume, find the custom requester.
3. Save the custom requester to the current folder.
4. Right-click the custom requester. In the context menu, choose Properties.
5. Choose Dependency.
6. In Dependency, choose Add to add the .rox file as a dependency for the custom requester.
7. Choose OK.

If there is more than one custom HTML requester form for a particular .rox file, ReportCast displays a form similar to the following when it receives a request to run a report.



Requester Menu

Please select from the following:

forcastreq2
forcastreq

Standard Requester

Copyright © 1999 Actuate Corporation

How to select a requester form

The user selects the appropriate requester form link to use that form.

Specifying parameters for report document output and version names

You can customize a requester form to use the report generation date as a parameter to assign a unique output name and version name to an e.report. Specify the report generation date by typing a date/time expression enclosed in braces, {}, in the Version Name and Output Name fields of the requester form.

The following date/time expression assigns the name Sales report as of the current date. For example, if the current date is March 1, 2001, the name of the e.report is Sales report as of 03-01-2001.

```
Sales report as of {mm-dd-yyyy}
```

ReportCast does not evaluate the date/time expression. The e.Reporting Server evaluates the date/time expression when ReportCast submits the report for generation.

See *Using e.Reports* for more information about specifying a date/time expression as a parameter for an e.report's output name and version name.

Reloading template files

Changes to the template files are not reflected in the generated HTML pages immediately. You must reload the templates from the ReportCast Administration page to see the changes. Only users with administrative privileges can access the ReportCast Administration page.

How to reload templates

1 To access the ReportCast Administration page, choose the Administration tab or type the URL in your browser.

The URL is similar to:

```
http://<webserver>/acweb/_ _admin.
```

where <webserver> is the name of the machine that runs your web server. If you choose the Administration tab, the ReportCast Channel Administration page appears. Click the ReportCast administration link on the left side of the page. The ReportCast administration page appears.

2 Choose Reload Templates.

The Confirmation page appears if the templates reload successfully.

3 Verify your changes by looking at the appropriate ReportCast page.

Naming template files

The following table lists the names of the ReportCast template files. All templates have the file extension .achtml. All templates are in the Standard directory of the web server's Actuate directory. You must follow this naming convention for any custom template files you create. Otherwise, ReportCast will not find them.

Template	Description
Active	Displays active requests.
Activedetail	Lists the properties for an active request.
Admin	Displays the ReportCast Administration page.
Cancel	Displays confirmation that a request has been canceled.
Channeladmin	Displays the ReportCast Channel Administration page.
Channelcontents	Displays the contents of a particular channel.
Channeldetail	Lists the properties for a particular channel.
Channelitemdetail	Lists the properties for a particular channel item.
Channelsubscription	Displays the ReportCast channels Subscription page.
Confirm	Displays confirmation that a request to run a report was submitted to e.Reporting Server.
Error	Displays information about an error condition.
Filedetail	Lists the properties for a particular report item.
Fileversions	Lists the available versions of the current file.
Folderlist	One-page view of a Report Encyclopedia folder.
Index	Custom folder page created by the administrator and stored directly on the e.Reporting Server.
Personalchcontents	Displays the contents of a particular user's Personal Channel folder.
Request	Requester page generated by ReportCast.
Requestdetail	Displays the details for a particular request.
Requestermenu	Lists the available requester forms. This page appears only if custom requester forms are available.
Requestlist	Lists the Active, Scheduled, or Completed requests.

Template	Description
Searchresults	Displays the results of searches on DHTML reports.
Scheduled	Lists all scheduled requests on the current e.Reporting Server.
Scheduleddetail	Lists the properties of a scheduled request.
Status	Displays the status of an active or completed request.
Success	Displays confirmation that a Drop or Rename operation completed successfully.
Viewframeset	Displays the frameset for viewing the DHTML report.
Viewnav	Composes the toolbar used for viewing DHTML reports.
Viewtoc	Displays the DHTML report’s table of contents.

The following table maps ReportCast directives to the template they generate or affect.

Directive	Template
_ _active	Requestlist
_ _completed	Requestlist
_ _scheduled	Requestlist
_ _channels?admin	Channeladmin
_ _channels?retrieveForm	Channelsubscription
_ _channels?submitForm	Confirm
_ _channels?retrieveCreation	Channeldetail
_ _channels?retrieveDetails	Channeldetail
_ _channels/channel?drop	Success
Cancel	Cancel
DeleteStatus	Success
Detail	Filedetail
Request	Requestermenu
Sort and Direction	Folderlist
StdRequest	Request
Submit	Confirm

Directive	Template
Drop	Cancel
FlushConnections	Admin
ViewFrameset	Viewframeset
GetDynamicData	Viewframeset
ViewPage	Viewframeset
GetReportData	
GetStyleSheet	Viewframeset
ViewTOC	Viewtoc
ViewNavigation	Viewnav
Versions	Fileversions
requestSearch	Searchrequest
submitSearch	Searchresults

Placing custom templates in a Report Encyclopedia

When you want to create a distinctive design for a particular department's e.Reporting Server web pages, use the Administrator Desktop's Navigator to place custom template files directly in the e.Reporting Server Encyclopedia. You must have administrator privileges on both the web server and the e.Reporting Server to perform this procedure.

How to place custom templates in a Report Encyclopedia

1 Create the _ _WebAgent folder on your Report server:

   1 Start Actuate Administrator Desktop.

   2 Choose File➛Navigator.

   3 In Report Encyclopedia Login, type your login information. Choose OK.

   4 Choose the e.Reporting Server destination for the custom ReportCast templates.

   5 Choose File➛New Encyclopedia Item➛Folder.

   6 In Create a new folder, in Enter folder name, type:

   ```
   _ _WebAgent
   ```

   7 Choose OK.

   The _ _WebAgent folder appears in the Navigator windows.

2. Create custom templates for the Report Encyclopedia:
   1. Copy the default templates from the web server configuration’s Standard directory to a work directory.
   2. Modify the templates.
   3. In Navigator, open the _ _WebAgent folder.
   4. Choose File➛New Encyclopedia Item➛Report Item.
   5. In Save to Volume, in Files of Type, choose All Files (*.*).
   6. Browse to the location of your custom templates.
   7. Select a template to place in the _ _WebAgent folder.
   8. Choose Save to Volume.

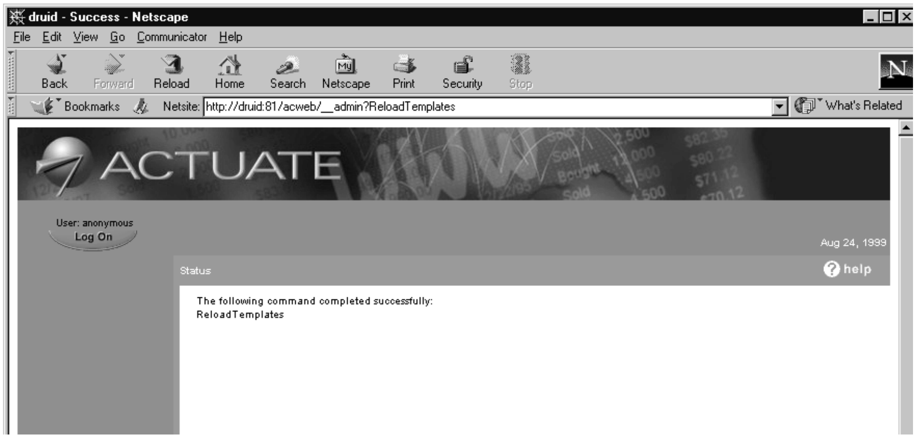
      The template appears in the Navigator window _ _WebAgent folder contents listing.

Repeat these steps for each template you want to place in the _ _WebAgent folder.

Viewing the HTML source for ReportCast templates

To view the HTML source of a page you see in the Netscape browser, choose View➛Document Source. The source document identifies the template file ReportCast used to generate the page.

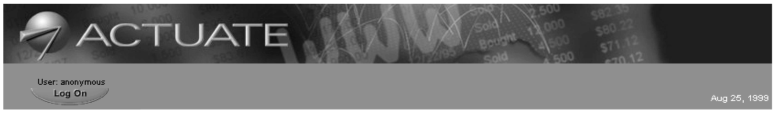
The following illustrations show the source for the Success.achtml template file and the web page it generates. The Success web page is similar to the following illustration.



The following command completed successfully:
ReloadTemplates

The following sections show the template source code that generates specific parts of the Success web page.

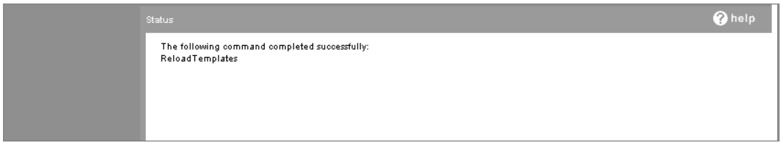
Web page title, background, banner, page header, user name, date, and sidebar graphics



User: anonymous

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<!-- Actuate Template File: $(TemplateName) -->
<!-- #Actuate Include "header.achtml" -->
<head>
<!-- #Actuate Include "browsertest.achtml" -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<Meta Name="Generator" Content="Actuate Web Agent 3.0">
<title>$(Host) - Success</title>
<!-- #Actuate Include "headerscript.achtml" -->
</head>
<!-- #Actuate Include "bodybegin.achtml" -->
<!-- #Actuate Include "banner.achtml" -->
<!-- PageHeader Begin -->
<!-- #Actuate Set CurrentTab = tabNone -->
<!-- #Actuate Set ServerString = "" -->
<!-- #Actuate Include "pageheader.achtml" -->
<!-- PageHeader End -->
<!-- SideBar Begin -->
<!-- #Actuate Include "sidebarbegin.achtml" -->
<!-- #Actuate Include "sidebarsuccess.achtml" -->
<!-- #Actuate Include "sidebarend.achtml" -->
<!-- SideBar End -->
```

Display Status, result of command execution, command executed



The following command completed successfully:
ReloadTemplates

```
<!-- #Actuate Include "contentheader.achtml" -->
<!-- PageTitle Begin -->
<!-- #Actuate Set PageTitle1 = "Status" -->
<!-- #Actuate Set PageTitle2 = "" -->
```

```
<!-- #Actuate Set PageTitle3 = "" -->
<!-- #Actuate Set PageTitle4 = "" -->
<!-- #Actuate Set PageTitle5 = "" -->
<!-- #Actuate Include "pagetitle.achtml" -->
<!-- PageTitle End -->
<!-- Content Begin -->
<!-- #Actuate Include "contentbegin.achtml" -->
<!-- #Actuate Include "contentsuccess.achtml" -->
<!-- #Actuate Include "contentend.achtml" -->
<!-- Content End -->
```

Display copyright information, page footer, close page elements

Copyright © 1997-1999 Actuate Corporation

```
<!-- #Actuate Include "contentfooter.achtml" -->
<!-- #Actuate Include "pagefooter.achtml" -->
<!-- #Actuate Include "bodyend.achtml" -->
<!-- #Actuate Include "metaend.achtml" -->
</html>
```

For more information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.”

Creating custom templates

This section provides examples of customizing the HTML pages generated by ReportCast.

How to customize a particular template

1. Display the standard web page for the template you want to customize.
2. In your web browser, go to a Report Encyclopedia folder containing report executables.
3. Choose a report executable.
4. In the Requester page in your web browser, choose File➛Save As.
5. Save the web page in your configuration’s Custom directory.
6. Edit the file in the Custom directory.

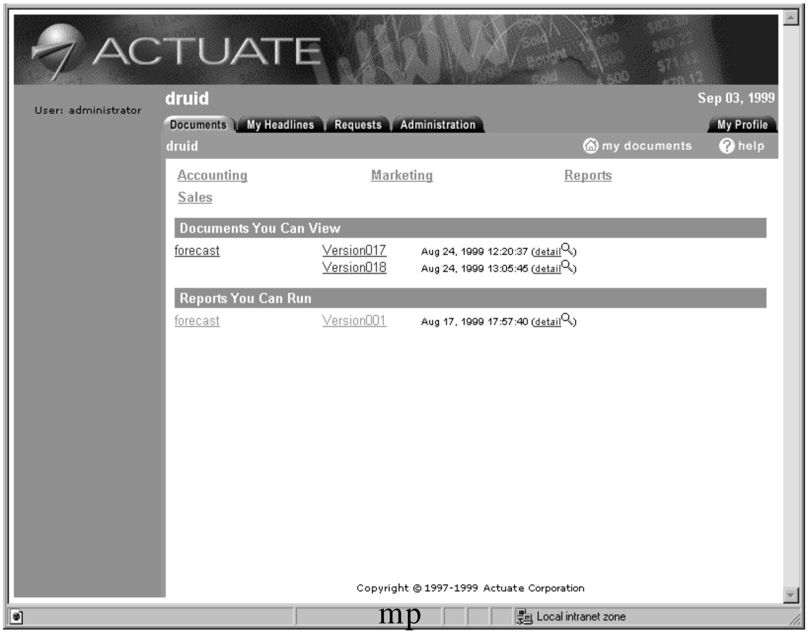
The standard templates include other template files and use variable substitution to display appropriate banners and icons on the resulting web page. The advantage to this method is that various elements are easily reused by several templates. The disadvantage is that elements such as nested included files can be confusing.

You can display the page you want to customize in the web browser, then save the page. The web page source is much more straightforward. Substitutions have been made, and the contents of included files are specifically included. You can edit and customize the complete web page.

Customized templates

This example shows how you can modify template files to change the format and data that ReportCast generates when it receives an http://.../<folder> directive. The following illustration shows the HTML pages generated from the default and custom template files.

Default HTML page generated from an http://.../<folder> directive



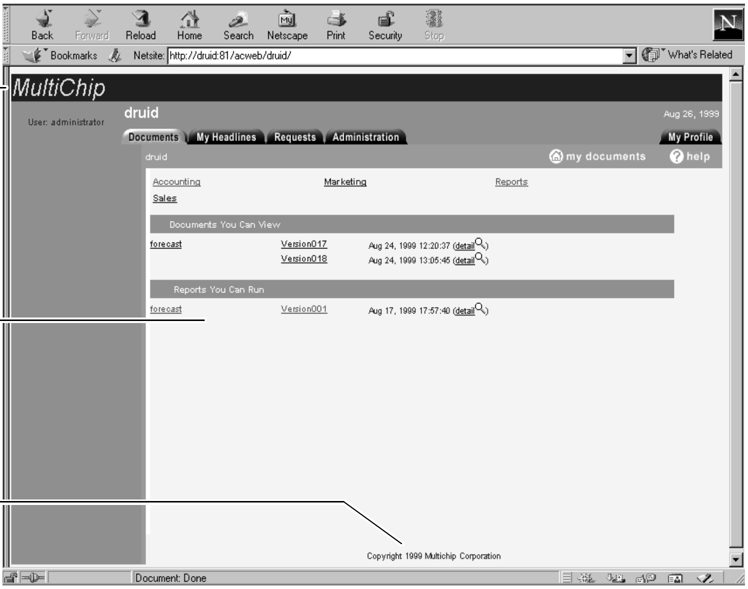
Accounting Marketing Reports
Sales

Documents You Can View		
forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run		
forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

Customized HTML page



Accounting Marketing Reports
Sales

Documents You Can View		
forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run		
forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright 1999 Multichip Corporation

Actuate logo replaced by MultiChip

Background color changed

Copyright changed to 1999, MultiChip

How to modify templates

To make the changes shown in this example, you modify these files:

- Folderlist.achtml, the template that generates Folder listing pages.
- Contentfolderlist.achtml, which Folderlist.achtml includes to generate the content for the Folder listing pages.
- Header.achtml, which Folderlist.achtml includes to define reporting web page colors, element names, commonly used URLs, and locations of images.
- Headerscript.achtml, a JavaScript file Folderlist.achtml includes to preload images for the reporting web page.
- Banner.achtml, which Folderlist.achtml includes to generate the company logo.
- Copyright.achtml, which Folderlist.achtml includes to generate the copyright.

Before you get started, create a directory called Custom in your configuration directory. For example, if you are using the U.S. English configuration, create the custom directory in your web server's \Actuate\Enu directory.

You must reload the templates to activate your changes. Reload templates from the ReportCast administration page, or use the ReloadTemplates directive.

How to replace one logo with another

Because the following is such a common requirement, the following steps take two sample logos and describe how to replace the Actuate logo with the Multichip logo:

1. Copy Banner.achtml from the Standard directory to the Custom directory.
2. Edit the file.

   The default contents are:

   ```
   <tr>
     <td valign="top" colspan=1 bgcolor="$(colorLogoBack)" nowrap>
         <img src="$(imgLogo1)" border="0" alt=""></td>
     <td width="100%" valign="top" colspan=6 bgcolor="$(colorLogoBack)"
         nowrap><img src="$(imgLogo2)" border=0 alt="Actuate Corporation">
     </td>
   </tr>
   ```

   The images imgLogo1 and imgLogo2 generate the default Actuate logo and banner for all the reporting web pages. The Header.achtml template defines imgLogo1 and imgLogo2.

3. Replace the HTML code:

   ```
   <img src="$(imgLogo1)" border="0" alt="">
   ```

   as follows:

   ```
   <font color=”yellow” size=”6” face=”Courier”><strong><em>
      MultiChip Corporation</em></strong></em></font>
   ```

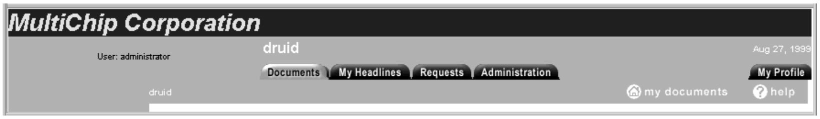
4. Replace the HTML code:

   ```
   <img src="$(imgLogo2)" border=0 alt="Actuate Corporation">
   ```

   as follows:

   ```
   &nbsp;
   ```

5. Save Banner.achtml.
6. View the changes:
   1. Reload the templates.
   2. Return to the ReportCast home page. The home page is now similar to the following image.



How to change the border color

1. Copy the templates Folderlist.achtml, Pageheaderx.achtml, and Header.achtml to the custom directory.
2. Edit Pageheaderx.achtml:
   1. Search for the string BGCOLOR.

      Search results include lines such as this one:

      ```
      <TD HEIGHT="10" COLSPAN="6" VALIGN="bottom"
      BGCOLOR="$(colorLightBlue)">
      ```

   2. Replace $(colorLightBlue) with turquoise. For example:

      ```
      <TD HEIGHT="10" COLSPAN="6" VALIGN="bottom"
      BGCOLOR="turquoise>
      ```

   3. Save Pageheaderx.achtml.
3. Edit Folderlist.achtml:
   1. Search for the string BGCOLOR.

      Search results include lines such as this one:

      ```
      <TD WIDTH="149" BGCOLOR="$(colorLightBlue)" VALIGN="TOP"
      NOWRAP><IMG SRC="$(imgSpace)" HEIGHT="5" WIDTH="149"></TD>
      ```

2 Replace $(colorLightBlue) with turquoise.

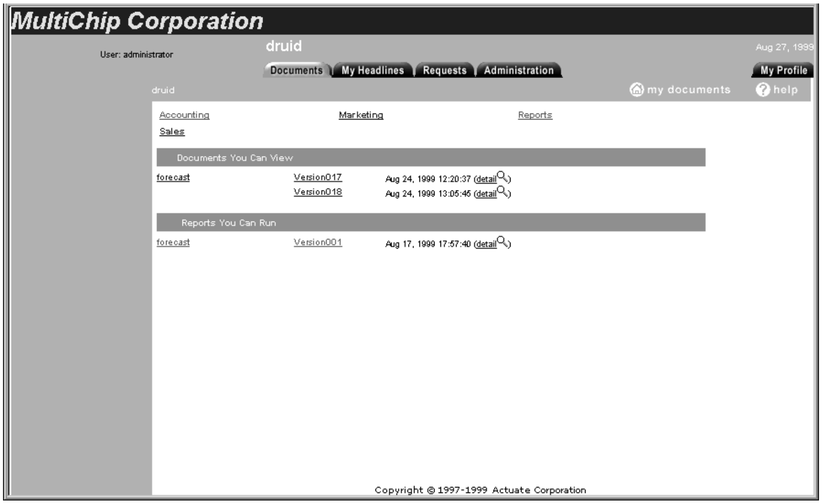
For example:

```
<TD WIDTH="149" BGCOLOR="turquoise" VALIGN="TOP" NOWRAP>
<IMG SRC="$(imgSpace)" HEIGHT="5" WIDTH="149"></TD>
```

4 Save Folderlist.achtml.

5 Reload the templates.

The web page looks similar to the following illustration.



Documents You Can View

forecast Version017 Aug 24, 1999 12:20:37 (detail)
Version018 Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast Version001 Aug 17, 1999 17:57:40 (detail)

The steps in the previous procedure result in a turquoise border for the Folder listing page. You can also include custom background images or set custom background colors for your Folder listing page.

The Header.achtml file contains settings for all the templates' image variables.

How to change the copyright information

The standard templates use a copyright image to display copyright information on the reporting web pages. To change the copyright information, change the image file, Img_copyright.gif, in the images directory.

Customizing the e.Analysis add-on option

The e.Analysis application is an add-on product that works with DHTML reports on the Web. You use e.Analysis to analyze data fields selected in a Smart Search. For more information, see *Using e.Analysis.*

If e.Analysis is installed, you can customize in Searchresults.achtml whether the Analyze Results button appears or whether Analysis appears as an item on the Download search results as: menu.

How to change where the user launches e.Analysis

To change where the user launches e.Analysis, you modify the Searchresults.achtml template. If your Searchresults.achtml template is in the custom directory of your default Actuate web reporting configuration directory, that template is the one you must modify. If your Searchresults.achtml template file is not in the custom directory of your default Actuate web reporting configuration directory, you must modify the Searchresults.achtml in the standard directory of your default Actuate web reporting configuration directory.

1 Using a text editor, open the Searchresults.achtml template file.

2 Find the following two variables in the Searchresults.achtml template file:

- var showAnalysis
- var showAnalysisButton

3 To change the visibility of e.Analysis, set the values of the variables based on the following table.

e.Analysis visibility	Statement in template file

To show Analyze Results button	var showAnalysis = true;
	var showAnalysisButton = true;

To show Analysis on drop-down menu	var showAnalysis = true;
	var showAnalysisButton = false;

To hide e.Analysis from user	var showAnalysis = false;
	var showAnalysisButton = true;

4 Reload the ReportCast templates:

```
http://<webserver>acweb<report server>_ _admin?ReloadTemplates
```

Where <webserver> is the name of your web server, and <report server> is the name of the e.Reporting Server. When users complete a Smart Search, the Smart Search page displays your configuration for the launch of e.Analysis.

How to enable launching e.Analysis

The e.Analysis install places the Searchresults.achtml template and the Searchresults.achtml.new template in the default configuration's Standard directory. If you have a modified Searchresults.achtml template in the Custom directory, the Analyze Results button does not appear on the Search Results page and you cannot launch e.Analysis.

To add the e.Analysis button to your modified Searchresults.achtml template:

1 Open the Searchresults.achtml.new template located in the <Wwwroot>\Actuate\Default\Standard directory.

This template contains the information you need to enable e.Analysis.

2. Open the Searchresults.achtml template located in the <Wwwroot>\Actuate\Default\Custom directory.
3. Add the code differences from the Searchresults.achtml.new in the Standard directory to the Searchresults.achtml template file in the Custom directory.
4. Find the variables showAnalysis and showAnalysisButton in the Custom directory's Searchresults.achtml template.
5. Set the values as follows:
   - var showAnalysis = true;
   - var showAnalysisButton = true;
6. Reload the ReportCast templates:

   ```
   http://<webserver>acweb<report server>_ _admin?ReloadTemplates
   ```

When users complete a Smart Search, the Smart Search page displays your configuration for the launch of e.Analysis.

Customizing the DHTML and SmartSearch displays

This section describes how to customize the DHTML and SmartSearch displays. The following table lists the files that affect DHTML and SmartSearch display:

File	Description
Browser type	Determine DHTML viewer behavior based on the web browser in use.
Browsertype.js	
Frameset	Determine the DHTML frameset look and feel
Viewframeset.achtml	
Toolbar	Determine the look and feel of the DHTML toolbar
Viewnav.achtml	
Viewnav.js	

File	Description
Search	Determine the SmartSearch look and feel
Searchrequest.achtml	
Searchresults.achtml	
Search.js	
Navbar (Table of Contents)	Determine the look and feel of the DHTML table of contents
Viewtoc.achtml	
Toctree.js	

Chapter

4

ReportCast directives

This chapter contains the following topics:

- About ReportCast directives
- ReportCast directives reference
- Examples of ReportCast directives

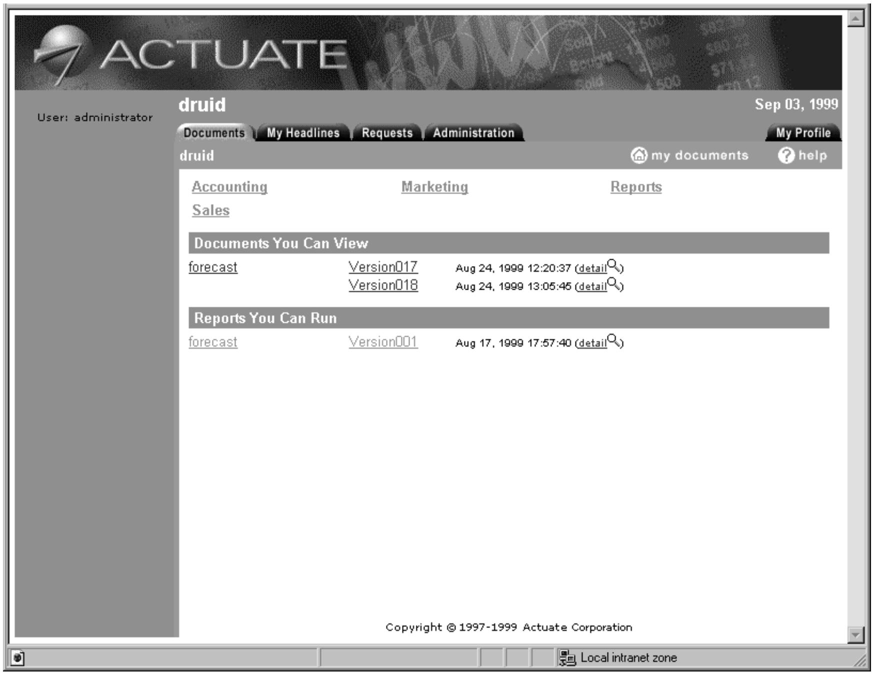
About ReportCast directives

ReportCast performs tasks based on directives from the web browser. Directives are URLs that contain the keyword acweb followed by the directive. When the web server encounters the acweb keyword, it passes the entire URL to ReportCast for processing.

Sending directives

Directives are often associated with buttons or links in the HTML pages generated by ReportCast. Choosing one of those buttons or links sends a directive. Another way to send a directive is to type the URLs in the Address or Location window of the web browser.

The following illustration shows a sample page generated by ReportCast. The page contains many links that, when clicked, send the appropriate URL directive to ReportCast.



User: administrator

druid

Sep 03, 1999

druid

Accounting Marketing Reports
Sales

Documents You Can View

forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

You can see what directive is associated with a link by placing the cursor over the link.

Managing directives

ReportCast uses HTML template files to generate HTML pages in response to directives sent from the web browser. Templates can be report-specific, folder-specific, e.Reporting Server-specific or web server-specific. ReportCast locates these template files by searching in a specific order:

1 The e.Reporting Server folder.

2 The e.Reporting Server's _ _WebAgent directory.

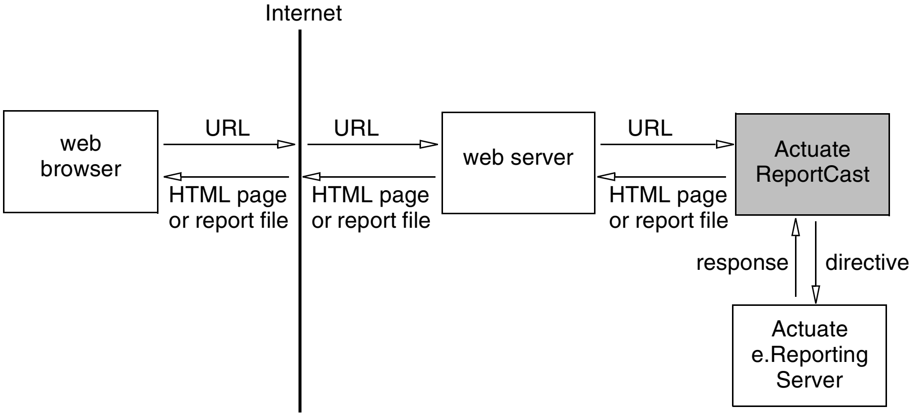
3 The web server's Custom directory for a particular configuration of the web server.

4 The web server's Standard directory.

For more detailed information about the search order, see "Understanding ReportCast searches for templates," in Chapter 3, "Working with ReportCast templates."

For detailed information about the HTML template files, see "About ReportCast templates," in Chapter 3, "Working with ReportCast templates."

The following illustration shows the flow of communication from the web browser to ReportCast and from ReportCast back to the web browser.



Formatting directives

ReportCast directives use one of the following formats:

{http | https}://<webserver>/acweb/[<configuration>/]<reportserver> | _ _local | _ _default/[<path>][<resource>][;<version>][?<command>]

or:

{http | https}://<webserver>/acweb/_ _admin

In the UNIX environment, for CGI programs only, ReportCast directives use one of the following formats:

```
{http | https}://<webserver>/<cgi script location>/acweb/[<configuration>
   ]<reportserver> | _ _local | _ _default/
   [<path>][<resource>][;<version>][?<command>]
```

or:

```
{http | https}://<webserver>/<cgi script location>/acweb/_ _admin
```

The following section provides additional information about each of the parameters of the directives.

Pararmeter

http | https

- Required.
- Specifies that the web browser and web server communicate using the HTTP communications protocol.

If your web server runs on the Secure Sockets Layer, URLs begin with https instead of http.

<webserver>

- Required.
- Name of the web server machine.

<cgi script location>

- Required to run ReportCast on UNIX using CGI programs.
- Not applicable to other platforms.

acweb

- Required.
- Tells the web server to pass the rest of this URL's fields to ReportCast.

_ _admin

- Optional.
- Directive that accesses the ReportCast Administration page.

<configuration>

- Optional.
- Name of the set of templates ReportCast should use to generate web pages for this request.

<reportserver> | _ _local | _ _default

- Required.
- Name of the e.Reporting Server you want to access.

If the e.Reporting Server, ReportCast Server, ReportCast Agent and web server are installed on the same machine, specify the literal _ _local instead. _ _local is the name of the web server machine that has connected to the ReportCast Server machine. This is the web server machine itself, not a proxy server. To resolve _ _local to a host name, DNS on the ReportCast Server machine must be able to find a host name for the address of the web server machine. Otherwise, _ _local is blank.

Specify the literal _ _default to access the e.Reporting Server on the machine specified in the AC_REPORTCAST_DEFAULT_REPORT_SERVER environment variable on UNIX, or registry key on NT.

ReportCast internally substitutes the e.Reporting Server's machine name for _ _local or _ _default. For example, if the user enters the following URL:

```
http://webserver/acweb/_ _local
```

ReportCast includes _ _local in all the links that it generates. The actual name of the machine displays in HTML. For example, ReportCast generates the following URL for a Report Encyclopedia folder called MyFolder:

```
http://webserver/acweb/__local/MyFolder/.
```

For information about setting the default e.Reporting Server, see "Setting ReportCast default e.Reporting Server," in "Administering ReportCast."

<path>

- Optional.
- Full path to the Report Encyclopedia folder or file to access.

<resource>

- Optional.
- A file or other resource in the Report Encyclopedia folder specified by <path>. If you do not specify this field, ReportCast accesses the folder specified by <path>.

;<version>

- Optional.
- Version number of the folder, file, or other resource to access.

?<command>

- Optional.
- Action to execute.

ReportCast directives reference

This section presents the types of ReportCast directives, and an alphabetical listing of directives and their syntax. Each directive entry includes a general description of the directive and a summary of its options.

Types of ReportCast directives

The following table lists the types of ReportCast directives.

Type of directive	Description
Folder directive	Specifies how the contents of a Report Encyclopedia display. Includes the Files and Sort and Direction directives.
File directive	Accesses and downloads report files from the e.Reporting Server. Includes the View directive.
Factory directive	Generates new reports from report executables (ROX files) or report parameter values (ROV). Factory directives also submit requests for execution. Includes Request, StdRequest, and Submit directives.
Status directive	Retrieves status about specific report-generation requests and about all report-generation requests handled by the factory on the specified e.Reporting Server. Includes the _ _active, _ _completed, _ _scheduled, DeleteStatus, Detail, and Drop directives.
Channel directive	Administers channels and retrieves channel contents. Includes the _ _channels?admin, _ _channels?retrieveForm, _ _channels?submitCreation, and _ _channels/<channel>?drop directives.
User preferences directive	Displays the user preferences form and submits user preferences changes. Includes the _ _requestUserPreferences and _ _submitUserPreferences directives.
Viewing directive	Determines how to view completed reports. Includes the View, ViewDefault, ViewFrameset, ViewPage, and ViewTOC directives.
Search directive	Performs searches on DHTML reports. Includes the requestSearch and searchReport directives.

Description of ReportCast directives

The following table lists the ReportCast directives and describes their functionality.

Directive	Description
_ _admin	Accesses the ReportCast administration page. Do not specify a report server in the URL when accessing the administration page.
_ _home	Accesses the current user’s home folder in the Report Encyclopedia, as set in the Administrator Desktop.
?DeleteStatus	Deletes status of a report-generation request.
?Detail	Generates detailed information about a Report Encyclopedia object or request.
?Drop	Cancels the report-generation request. Deletes a Report Encyclopedia object.
?Files	Lists only the files the current folder contains. Does not list subfolders.
?Reports	Displays the reports in the specified folder, sorted by name.
<report URL>	Downloads the entire report document.
?Request	Generates requester forms.
?StdRequest	Generates the standard requester form.
?Submit	Submits a request. Starts the report generation process.
?Status	Displays status for the current report-generation request.
?Sort and Direction	Specifies how to display the contents of a Report Encyclopedia.
Versions	Displays a list of the available versions of a file when the file the user is viewing in DHTML becomes unavailable.
?View	Displays a report, one page at a time, in a browser. For HTML reports,?View displays the entire report in the browser.
<report URL> ?ViewDefault	Displays a report in a format determined by the user’s preference and viewing privileges.
<report URL> ?ViewFrameset	Displays the report in DHTML.

Directive	Description
_ _scheduled	Displays the list of currently scheduled factory requests. Associated with the Requests tab on the e.Reporting web site.
_ _active	Displays the list of currently active factory requests. Associated with the Active link on the e.Reporting web site's Requests page.
_ _completed	Displays the list of completed factory requests and channels to which the user subscribes. Associated with the My Headlines tab on the e.Reporting web site.
_ _channels?admin	Displays the channels administration page.
_ _channels?retrieveForm	Displays the subscription form for available channels.
_ _channels?submitForm	Submits subscription information.
_ _channels?retrieveCreation	Displays the form used to add new channels.
_ _channels?submitCreation	Submits the new channel form.
_ _channels/<channel>?drop	Removes an existing channel.
_ _channels/<channel>?retrieveDetails	Displays a form used to modify channel attributes.
_ _channels/<channel>?submitDetails	Submits the channel details form.
_ _requestUser Preferences	Requests the user preferences page.
_ _submitUser Preferences	Submits the user preferences page as an HTML form.
<report URL>?GetDynamic Data	Retrieves data for a specific component, such as a graph or image in a report.
<report URL>?ViewPage	Retrieves a specific page of a report, either from the navigation bar or from a set of search results.
<report URL>?GetReportData	Retrieves report data in a particular format.

Directive	Description
<report URL>?GetStyleSheet	Retrieves the style sheet for the report in a particular format, such as CSS or XMLStyle.
<report URL>?ViewTOC	Retrieves the report's table of contents from the view process.
<report URL>?ViewNavigation	Displays the navigation bar.
<report URL>?Versions	Displays the list of available versions of a file, when the file the user is viewing in DHTML becomes unavailable.
<report URL>?searchReport	Performs a search on a DHTML report according to criteria that you specify in a URL that you embed in a report or web page.
<report URL>?requestSearch	Displays the search request page.
<report URL>?submitSearch	Submits the search request and returns the search results page.
<report URL>?extractSearchResults	Submits the search and returns search results in a specified format, such as comma-separated values or tab-separated values.

<Report URL>

Download a report.

Syntax http://<webserver>/acweb/<reportserver>/<folder>/<file>[?download]

Description Send this directive to download an entire report file from the e.Reporting Server.

If you configured the web browser to run the Actuate LRX or if you chose Live Report Extension as your report viewing preference, and you are downloading an Actuate report document (ROI), include the download keyword in your download URL. Including the download keyword causes the user's web browser to display a download dialog, allowing the user to save the report to their machine. The user later views the report using the LRX. For example, the following URL allows the user to download the detail report to the their machine for later viewing in the LRX.

```
http://xanadu/acweb/paradise/reports/detail.roi?download
```

If you chose Dynamic HTML (DHTML) as your report viewing preference, ReportCast does not download the report. The report displays in your web browser in DHTML format.

Downloading report document pages on demand

To view a report through the LRX, you can download an entire report document, or download only the pages you need. For large reports, the second method, called demand paging, saves time and network resources. For more information about demand paging, see the View directive later in this chapter.

Specifying file versions

When you issue a directive to download a file, you can also specify a specific version number. For example, the following directive downloads version 2 of Detail.roi in the Reports folder on the Paradise e.Reporting Server:

```
http://xanadu/acweb/paradise/reports/detail.roi;2
```

If a folder contains multiple versions of a report and you do not specify a version number, ReportCast downloads the latest version. If you specify a version number that does not exist, ReportCast displays an error message.

See also View

_ _active

Display the list of currently active requests.

Syntax http://<webserver>/acweb/<reportserver>/_ _active

Description The _ _active directive retrieves the list of requests that the factory is currently running and displays them in the web browser. Use with the status directives _ _completed and _ _scheduled to track requests.

Example

```
http://sales/acweb/north/_ _active
```

See also _ _completed
_ _scheduled

_ _admin

Perform ReportCast administration tasks. Available only to users with administrative privileges.

Syntax http://<webserver>/acweb/_ _admin[?FlushConnections | ?ReloadTemplates]

Description The _ _admin directive supports users with administrative privileges accessing the ReportCast administration page. ReportCast administration tasks include closing open e.Reporting Server connections and reloading the ReportCast templates. Use FlushConnections to close open e.Reporting Server connections. Use ReloadTemplates to reload the ReportCast templates. You must reload the ReportCast templates after modifying them. ReportCast does not recognize any changes until you reload the templates.

You cannot administer the e.Reporting Server from the ReportCast administration page.

Example The following _ _admin directive displays the ReportCast administration page.

```
http://sales/acweb/_ _admin
```

The following _ _admin directive reloads the ReportCast templates.

```
http://sales/acweb/_ _admin?ReloadTemplates
```

_ _channels

Administer, view contents of, subscribe to, and unsubscribe from ReportCast channels.

Syntax

```
_ _channels?admin

_ _channels?retrieveCreation

_ _channels?retrieveForm

_ _channels?submitCreation[&_ _ChannelName=<channel name>]
   [&_ _Comment=<comment>][&_ _Expiration=<expiration>]
   [&_ _LargeIconURL=<large icon url>][&PollingInterval=<interval>]
   [&_ _Roles=<roles>][&_ _SmallIconURL=<small icon url>]

_ _channels?submitForm[&channels=<channel name>]

_ _channels/<channel>?drop

_ _channels/<channel>?retrieveDetails

_ _channels/<channel>?submitDetails[&_ _ChannelName=<channel name>]
   [&_ _Comment=<comment>][&_ _Expiration=<expiration>]
   [&_ _LargeIconURL=<large icon url>][&PollingInterval=<interval>]
   [&_ _Roles=<roles>][&_ _SmallIconURL=<small icon url>]
```

Description Use channel directives to administer, view the contents of, subscribe to, and unsubscribe from ReportCast channels. For more information about channel directives, see “Working with ReportCast channels,” in Chapter 1, “Using ReportCast.”

The following table lists the channel directives.

Directive	Description
_ _channels?admin	Displays the channel administration page.
_ _channels?retrieveForm	Displays the subscription form for available channels.
_ _channels?submitForm	Submits the subscription information.
_ _channels?retrieveCreation	Displays a form to create a new channel. Available only to users with administrative privileges.
_ _channels?submitCreation	Submits the new channel creation form. Available only to users with administrative privileges.
_ _channels/<channel>	Lists the contents of channel <channel>.
_ _channels/<channel>?drop	Removes the existing channel <channel>. Available only to users with administrative privileges.
_ _channels/<channel> ?retrieveDetails	Displays channel attributes modification form for <channel>. Available only to users with administrative privileges.
_ _channels/<channel> ?submitDetails	Submits the channel details modification form for <channel>. Available only to users with administrative privileges.

Include channel variables in a submitCreation or submitDetails URL directive to administer channels without using the channel detail page form. The channel details page contains the following variables for administering channels:

Variable	Description
_ _ChannelName	Required for creating channels. A string containing the new channel's name.
_ _Comment	Optional. A string containing a description of the channel.

Variable	Description
_ _Expiration	Required. A number in locale-specific format specifying the number of days after which items expire from the channel.
_ _LargeIconURL	Optional. A URL to a 32x32 pixel icon representing the channel.
_ _PollingInterval	Required. A number in local-specific format specifying the frequency, in seconds, that the page listing the contents of this channel should refresh itself while a user views it.
_ _Roles	Optional. The name of a role on the e.Reporting Server that has permission to subscribe to this channel. Since many roles may subscribe to a channel, you can assign this variable more than once.
_ _SmallIconURL	Optional. A URL to a 20x20 pixel icon representing the channel.

Include the channel subscription variable, channels, in a submitForm URL directive to subscribe a user to a channel.

The _ _channels?retrieveForm directive lists available channels and allows users to subscribe or unsubscribe to channels of interest. Users see only the channels for which they have access permissions.

The _ _completed directive lists all completed requests as well as the channels to which the user subscribes. For more information about listing completed requests and channels, see _ _completed later in this chapter.

Example The following channels directive creates a new channel, News. The channel's description is All the news that fits. Note that the characters %20 represent spaces. Items expire from the channel in 5 days. The channel updates every 2 minutes. It is open to all users.

```
http://sales/acweb/north/_ _channels?submitCreation&
_ _ChannelName=News&_ _Comment=All%20the%20news%20that%20fits&
_ _Expiration=5&_ _PollingInterval=120
```

The following channels directive modifies the News channel so that items expire in 10 days.

```
http://sales/acweb/north/_ _channels/News?submitDetails&_ _Expiration=10
```

The following channels directive subscribes the current user to the News channel.

```
http://sales/acweb/north/_ _channels?submitForm&channels=News
```

The following channels directive lists the contents of the Sales channel.

```
http://sales/acweb/north/_ _channels/Sales
```

The following channels directive displays the channel details page for the News channel. The channel details page allows users with administrative privileges to modify channels.

```
http://sales/acweb/north/_ _channels/News?retrieveDetails
```

The following channels directive allows a user to subscribe to available channels.

```
http://sales/acweb/north/_ _channels?retrieveForm
```

The following channels directive allows users with administrative privileges to remove a channel.

```
http://sales/acweb/north/_ _channels?drop
```

See also _ _completed

_ _completed

Display the list of completed requests and the ReportCast channels to which a user subscribes.

Syntax http://<webserver>/acweb/<reportserver>_ _completed

Description The _ _completed directive displays the contents of a user's completed folder and the ReportCast channels to which the user subscribes. Users click the My Headlines tab or the Personal Channel link on the reporting web site to display their completed requests and the channels to which they subscribe.

Example The following _ _completed directive displays the contents of the user's completed folder on the North e.Reporting Server.

```
http://sales/acweb/north/_ _completed
```

See also _ _channels
_ _scheduled

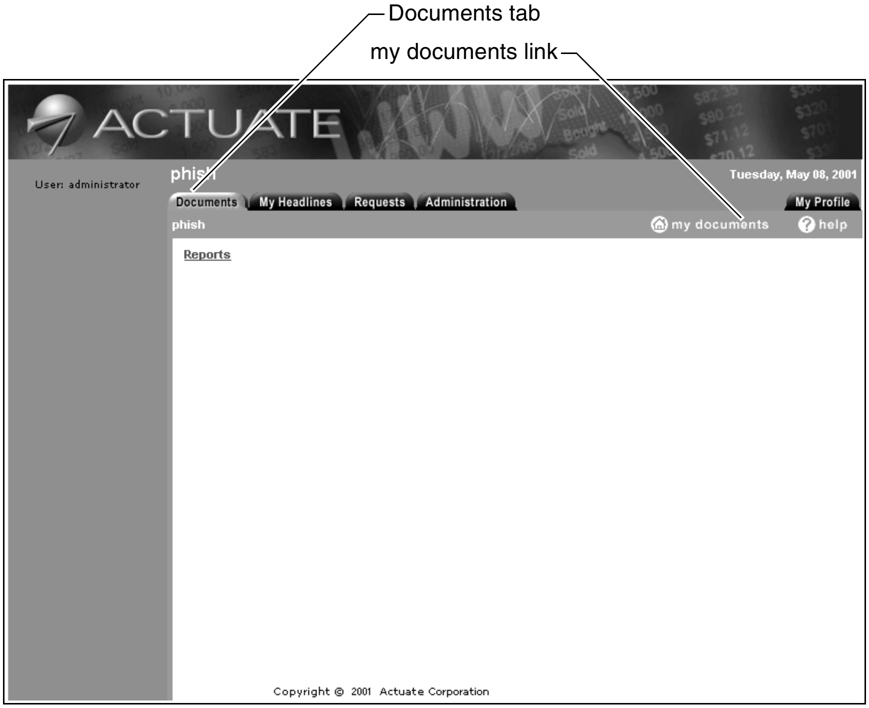
_ _home

Access the user's home folder as set in the Administrator Desktop.

Syntax http://<webserver>/acweb/<reportserver>_ _home

Description The _ _home directive displays the contents of the user's Report Encyclopedia home folder. Users with administrative privileges set the user's home folder in the Administrator Desktop. The default home folder is the reporting web site home page, http://<webserver>/acweb/<reportserver>.

Users click the Documents tab or the my documents link on the reporting web site to display their home page. The following illustration shows the Documents tab and the my documents link.



Example The following _ _home directive displays the user's home folder on the North e.Reporting Server.

```
http://sales/acweb/north/_ _home
```

_ _scheduled

Display the list of currently scheduled requests.

Syntax http://<webserver>/acweb/<reportserver>_ _scheduled

Description The _ _scheduled directive displays a list of all the currently scheduled requests on the specified e.Reporting Server. Users click the reporting web site's Requests tab to display the list of scheduled requests.

Example The following _ _scheduled directive displays the currently scheduled requests on the North e.Reporting Server.

```
http://sales/acweb/north/_ _scheduled
```

See also _ _active
_ _completed

_ _requestUserPreferences

Display the User Preferences page as an HTML form.

Syntax

```
http://<webserver>/acweb/<reportserver>_ _requestUserPreferences
```

Description The _ _requestUserPreferences directive displays the User Preferences page as an HTML form. The User Preferences page provides a way to make changes to users' personal profiles, including password changes, report viewing preference, or downloading an Actuate LRX Viewer.

Users access the User Preferences page by clicking the My Profile tab on the reporting web site. Users perform the following tasks to change user preferences:

- To change passwords, fill out the Change Password section of the form then click the Change Password button.
- To change reporting viewing preferences, choose a viewing preference in the Report Viewing Preference section of the form then click the Change Preference button.
- To download an Actuate LRX Viewer, click the URL for the desired version of the LRX.

Example The following _ _requestUserPreferences directive displays the user's Personal Preferences page.

```
http://sales/acweb/north/_ _requestUserPreferences
```

See also _ _submitUserPreferences

ViewDefault

_ _submitUserPreferences

Submit the User Preferences page as an HTML form.

Syntax

```
http://<webserver>/acweb/<reportserver>_ _submitUserPreferences
   [&ViewFormatPreferences=”DHTML” | “ROI” | “SYSTEM DEFAULT”]
```

Description The _ _submitUserPreferences directive submits the User Preferences page as an HTML form. User Preferences provides a way to make changes to users' personal profiles, including password changes, report viewing preference, or downloading an Actuate LRX Viewer.

Users access User Preferences by clicking My Profile on the reporting web site.

Example The following _ _submitUserPreferences directive submits the user's changes to their personal profile to ReportCast.

```
http://sales/acweb/north/_ _submitUserPreferences
```

See also _ _requestUserPreferences

DeleteStatus

Delete a report generation status.

Syntax

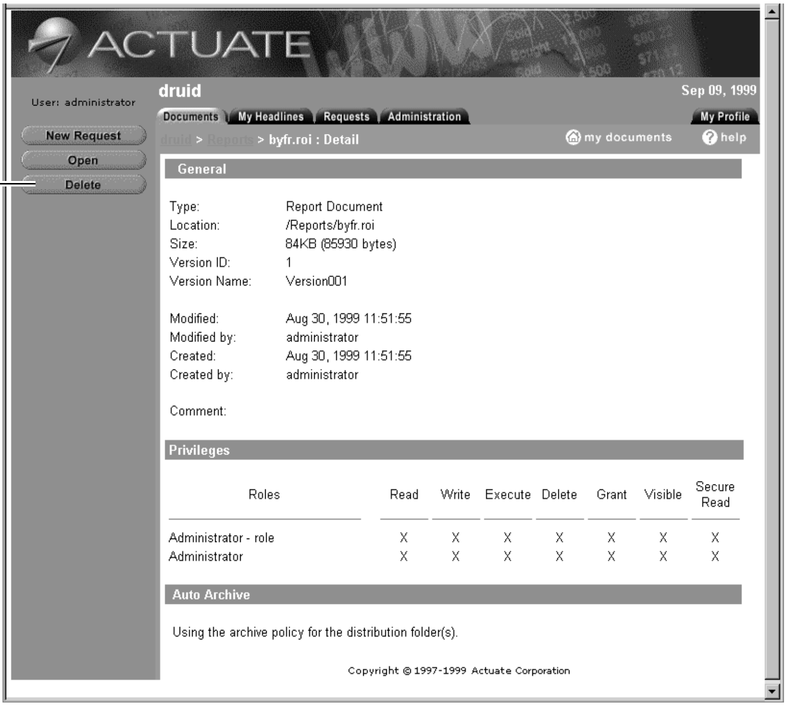
```
http://<webserver>/acweb/<reportserver>/_ _requests/
    <request id>?DeleteStatus
```

Description The DeleteStatus directive deletes any status the current user has for the request specified by <request id>. ReportCast creates status information automatically as a report generates.

For example, if you generate a particular report or set of reports, ReportCast creates status for each of them. The status is available from a report's details page. When you view the list of completed requests on your report server, you access the status by clicking the report's Details link. The Details page contains a Delete Notice button. Click Delete Notice to delete the status for that report.

The following illustration shows Details and Delete Notice.

Clicking Delete generates the ?Delete Status directive.



General

Type:	Report Document
Location:	/Reports/byfr.roi
Size:	84KB (85930 bytes)
Version ID:	1
Version Name:	Version001
Modified:	Aug 30, 1999 11:51:55
Modified by:	administrator
Created:	Aug 30, 1999 11:51:55
Created by:	administrator
Comment:	

Privileges

Roles	Read	Write	Execute	Delete	Grant	Visible	Secure Read
Administrator - role	X	X	X	X	X	X	X
Administrator	X	X	X	X	X	X	X

Auto Archive

Using the archive policy for the distribution folder(s).

Copyright © 1997-1999 Actuate Corporation

See also Detail

Detail

Display detailed information about a Report Encyclopedia object or about a report generation request.

Syntax Retrieve details about Report Encyclopedia objects:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?detail
   [&Page=Scheduled]
```

Retrieve details about report generation requests:

```
http://<webserver>/acweb/<reportserver>/_ _requests/<request id>?detail
```

Description The detail directive provides precise information about a Report Encyclopedia object or report generation request. The information provided is usually too extensive and too complex to display in ReportCast's summary views. The information depends upon the object type. For example, details about a folder include:

- The folder name
- The last modification date
- The owner
- The number of files and folder that the current folder contains

In general, information provided for Report Encyclopedia objects includes:

- All operations allowed on that object. Operations include:
    - Download
    - Drop
    - View
    - Request
- User privileges for accessing the object

ReportCast uses a format similar to the Navigator's property pages format to display the information. For more information about the Navigator, see Chapter 1, "Understanding the Administrator Desktop," in *Administering the Report Encyclopedia.*

The _ _requests/<request id>?detail directive displays a page that shows detailed status information about the report generation request. Users retrieve this information by clicking a link, such as the Request Status button on the Request Confirmation page.

Example The following example displays the details for a report request with ID 2001-05-07%2017%3A57%3A14%202. The character string %20 represents a space. The character %3A represents a colon (:). The ID is a date/time stamp.

```
http://sales/acweb/north/
_ _requests/ID%3A%202001-05-07%2017%3A57%3A14%202?detail
```

The following example displays the details for a scheduled request with ID 2001-05-07%2017%3A57%3A14%202. The character string %20 represents a space. The character %3A represents a colon (:). The ID is a date/time stamp.

```
http://phish/acweb/phish/
_ _requests/ID%3A%202001-05-07%2017%3A57%3A14%202?detail
&Page=Scheduled
```

See also _ _active
_ _completed
_ _scheduled

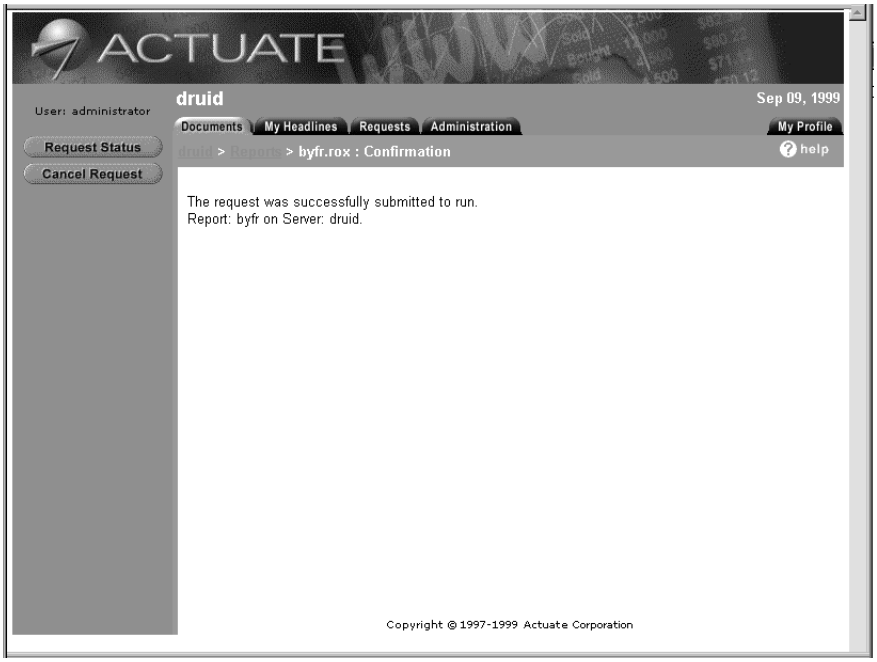
Drop

Cancel a report generation request.

Syntax http://<webserver>/acweb/<reportserver>/_ _requests/<request id>?drop

Description The drop directive cancels the request for status of a report-generation request just submitted. Users generally cancel report generation requests by clicking Cancel Request on the Request Successfully Submitted page.

The following illustration shows a status page for a report-generation request. Click Request Status to get more status information. Click Cancel Request to cancel the report-generation job.



The request was successfully submitted to run.
Report: byfr on Server: druid.

Copyright © 1997-1999 Actuate Corporation

Example The following drop directive cancels a report execution request.

```
http://sales/acweb/north/_ _requests/
ID%3A%202001-05-08%2017%3A49%3A37%202?Drop
```

See also _ _active
_ _scheduled
Request and StdRequest

extractSearchResults

Submit a search and return the results in a specified format.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
    ?extractSearchResults[&format=<format>]
```

Pararmeter

format=<format>
Specifies the format in which to return search results. Supported values for format are:

- CSV: comma separated values
- TSV: tab separated values

Description

The extractSearchResults directive submits a search request and returns the search results in the specified format. The search criteria are included in the search form data. HTTP GET and HTTP POST methods are supported. The search form data does not have to be contained in the URL.

Searches are case-sensitive.

For information about designing URL’s to link to a specific location in a report, see Chapter 10, “Programming for report viewing events,” in *Programming e.Reports*.

Example

The following extractSearchResults directive example submits a search request for the Marketing plans report and returns the results in TSV format. Assume that the user filled out an HTML form to specify the search criteria.

```
http://sales/acweb/north/Marketing/plans.roi
?extractSearchResults&format=TSV
```

See also

requestSearch
searchReport
submitSearch

Files

List the files in the current folder.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/?files
```

Description

The Files directive lists the files in the current folder. The Files directive does not list subfolders. You can combine the Files directive with the Sort and Direction directives to determine how the listed files appear.

Displaying only report files in a folder

The following table lists the directives you can send if you want to view only report files, not subfolders.

Directive	Displays ...
http://<webserver>/acweb/<reportserver>/<folder>/?files	Reports in the specified folder
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Name	Reports in the specified folder, sorted by name
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Type	Reports in the specified folder, sorted by file type
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Owner	Reports in the specified folder, sorted by creator
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Date	Reports in the specified folder, sorted by modified date
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Created	Reports in the specified folder, sorted by creation date
http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Size	Reports in the specified folder, sorted by size
http://<webserver>acweb/<reportserver>/<folder>?files&sort=Date&direction=desc	Reports in the specified folder, sorted by creation date in descending order

Example The following Files directive lists only the files that the Marketing folder contains.

```
http://sales/acweb/north/Marketing?files
```

The following Files directive lists only the files in the Marketing folder. The directive sorts the files by owner.

```
http://sales/acweb/north/Marketing?files&sort=owner
```

See also Sort and direction

GetDynamicData

Retrieve data for a specific component.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder><file>
    ?GetDynamicData&componentID=<ID>&scalingFactor=<scale>
```

Pararmeter

ComponentID=<ID>
Specifies the ID of the dynamically-generated component.

scalingFactor=<scale>
Specifies the size of the component or the zooming percentage. A scaling factor of 100 indicates the original size of the component. A scaling factor of 50 displays the component at 50% of its original size. 100 is the default scaling factor.

Description

The GetDynamicData directive retrieves data for a component that the view process generates dynamically. The component is usually a graph or image in a report.

See also

GetReportData

GetReportData

Retrieve report data in a specified format.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
    ?GetReportData&format=<format>[&componentID=<ID>]
    &scalingFactor=<scale>
```

Pararmeter

format=<format>
Specifies the format in which to display the report. Format is any format that the view process supports, such as XML structure or page-view formats, or DHTML. The valid values for format are:

- DHTML—a compressed DHTML format that uses Cascading Style Sheets (CSS). This is the default format.
- DHTMLLong—an uncompressed DHTML format. Use this format if your browsers do not support CSS.

componentID=<ID>
Specifies the ID of the dynamically-generated component.

scalingFactor=<scale>
Specifies the size of the component or the zooming percentage. A scaling factor of 100 indicates the original size of the component. A scaling factor of 50 displays the component at 50% of its original size. 100 is the default scaling factor.

operation=<operation>
Specifies whether to view the report in the browser or save the report in PDF format for printing. Valid values are:

- view—view the report in the browser
- print—save the report in PDF format for printing

Description The GetReportData directive retrieves data for a report in a specified format. If you do not specify a componentID, Actuate retrieves the entire report.

Example The following GetReportData directive retrieves the Marketing plans report in DHTML format at half its original size.

```
http://sales/acweb/north/Marketing/plans.roi
&format=DHTML&scalingFactor=50
```

See also GetDynamicData

GetStyleSheet

Retrieve the style sheet for a report in a specified format.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
   ?GetStyleSheet&format=<format>
```

Pararmeter **format=<format>**
Specifies the format in which to retrieve the style sheet. Valid values are CSS, for cascading style sheets for DHTML output, and XMLStyle, for style information specified in XML.

Description The GetStyleSheet directive retrieves the style sheet for the report in the specified format.

Example The following GetStyleSheet directive retrieves the style sheet for the Marketing plans report in XML format.

```
http://sales/acweb/north/Marketing/plans.roi?
GetStyleSheet&format=XMLStyle
```

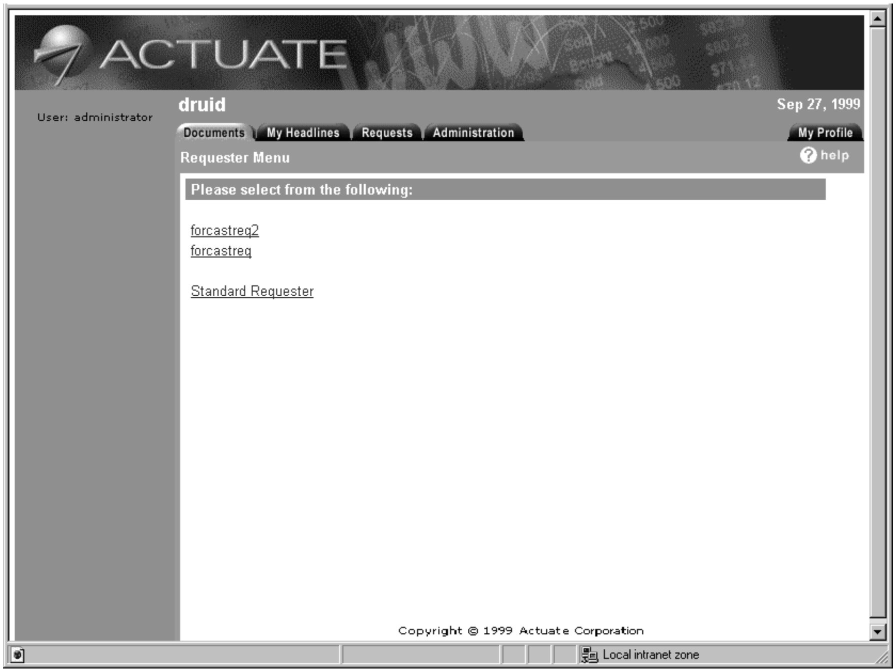
Request and StdRequest

Generate a requester form.

Syntax http://<webserver>/acweb/<reportserver>/<folder>/<.rox | .roi | .rov | .row>?Request | ?StdRequest

Description The Request and StdRequest directives generate requester forms that run reports. The requester forms display in the web browser. Users fill out the requester forms with request parameters, then click Submit Request to generate a Submit URL directive, which submits the request to the e.Reporting Server.

Use the Request directive to generate custom requester forms. Use the StdRequest directive to generate the default requester form. If more than one requester form is available, ReportCast displays a list of the forms to choose from, as shown in the following illustration.



Please select from the following:

forcastreq2
forcastreq

Standard Requester

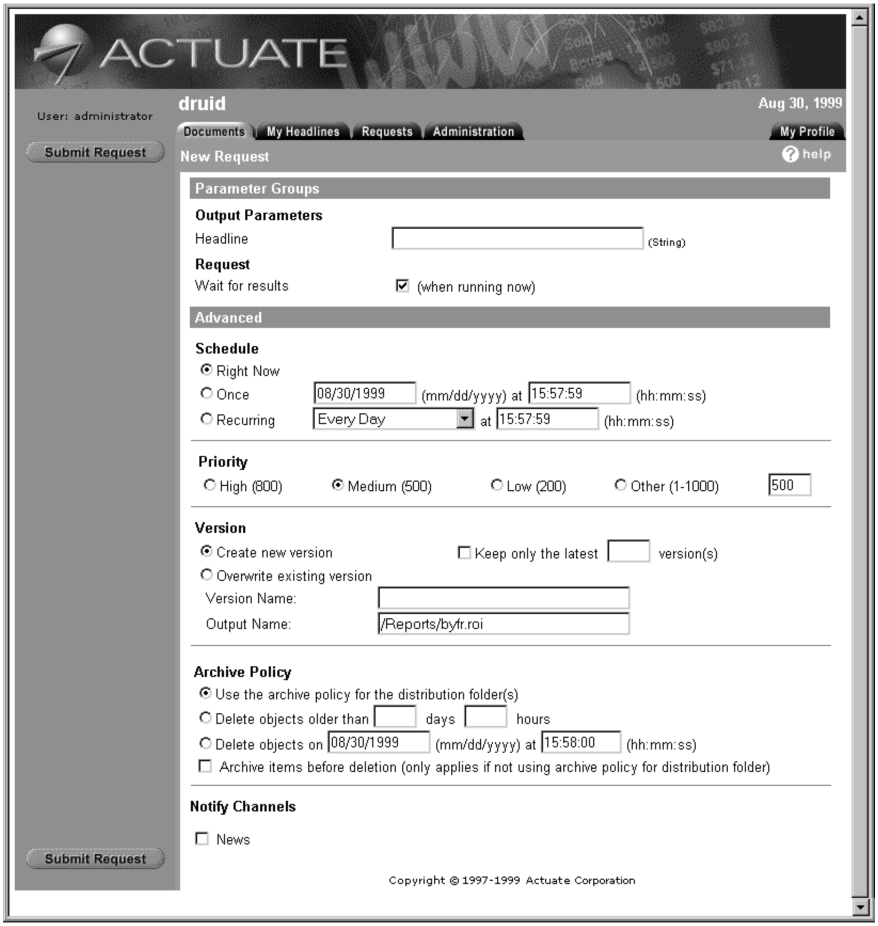
Copyright © 1999 Actuate Corporation

Custom requester forms make running reports simpler for your users. For example, suppose that your regional sales departments run a particular generic report on a regular basis. If you use a standard request, each time the sales staff runs the report, they have to fill in the region-specific information on the form. That information is always the same, so filling it in each time can be tedious. You can create a requester form for each region that supplies the appropriate parameter values for that region.

You can create as many custom requester forms as you need, but there must always be a default requester form for ReportCast to use. The type of requester form ReportCast generates depends on the type of file.

Report file type	Form
ROX	Generates a blank requester form from the ROX file parameter definitions.
ROV	Generates a blank requester form using the ROV file values as defaults.
ROI	Looks for the ROV file and uses it to generate a blank requester form. If ReportCast cannot find an ROV file, it uses an ROX file.
ROW	Looks for the ROV file and uses it to generate a blank requester form. If ReportCast cannot find an ROV file, it uses an ROX file.
HTML	Generates a custom HTML requester form for a particular report, depending on a specific ROX or ROV file.

The following illustration shows the standard requester form for the Byfr report. The Byfr report is in your Actuate5\ErdPro\Examples\FirstRpt directory.



Parameter Groups

Output Parameters

Headline [ ] (String)

Request

Wait for results ☑ (when running now)

Advanced

Schedule

● Right Now

○ Once 08/30/1999 (mm/dd/yyyy) at 15:57:59 (hh:mm:ss)

○ Recurring Every Day at 15:57:59 (hh:mm:ss)

Priority

○ High (800) ● Medium (500) ○ Low (200) ○ Other (1-1000) 500

Version

● Create new version □ Keep only the latest [ ] version(s)

○ Overwrite existing version

Version Name: [ ]

Output Name: /Reports/byfr.roi

Archive Policy

● Use the archive policy for the distribution folder(s)

○ Delete objects older than [ ] days [ ] hours

○ Delete objects on 08/30/1999 (mm/dd/yyyy) at 15:58:00 (hh:mm:ss)

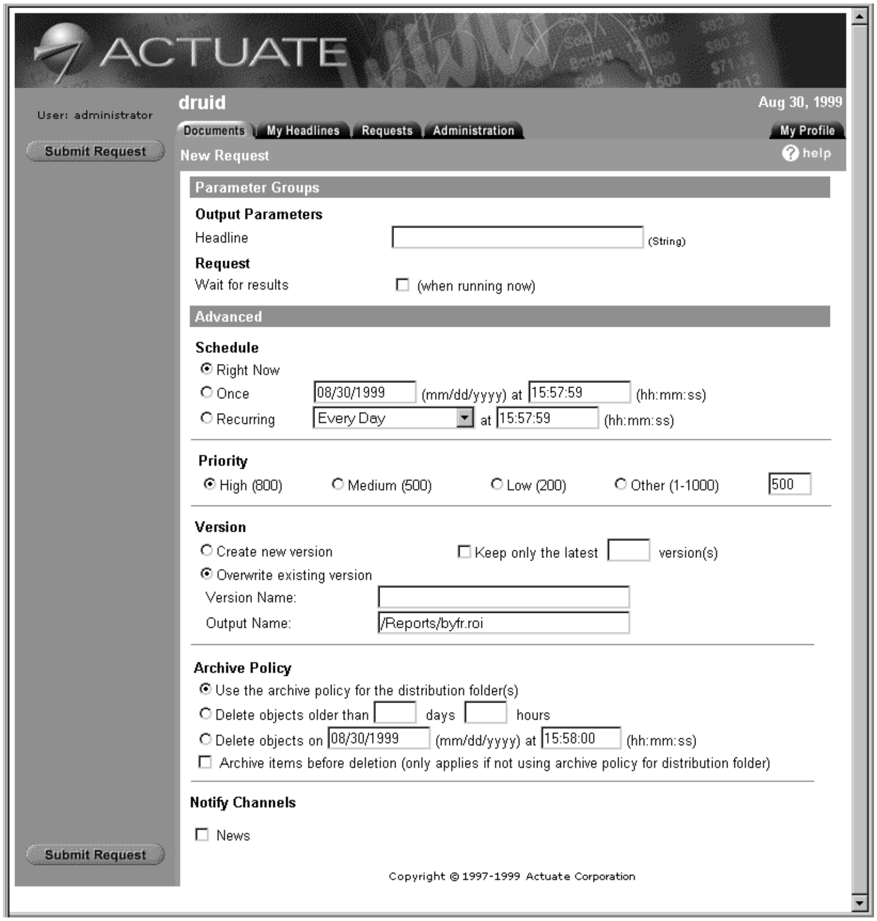
□ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels

□ News

Copyright © 1997-1999 Actuate Corporation

The following illustration shows the VIP custom requester form for the Byfr report. The VIP requester does not wait for the report to finish, runs the report at the highest priority level by default, and overwrites the existing report version rather than creating new versions each time.



Parameter Groups

Output Parameters

Headline [   ] (String)

Request

Wait for results □ (when running now)

Advanced

Schedule

● Right Now

○ Once 08/30/1999 (mm/dd/yyyy) at 15:57:59 (hh:mm:ss)

○ Recurring Every Day ▾ at 15:57:59 (hh:mm:ss)

Priority

● High (800) ○ Medium (500) ○ Low (200) ○ Other (1-1000) 500

Version

○ Create new version □ Keep only the latest [   ] version(s)

● Overwrite existing version

Version Name: [   ]

Output Name: /Reports/byfr.roi

Archive Policy

● Use the archive policy for the distribution folder(s)

○ Delete objects older than [   ] days [   ] hours

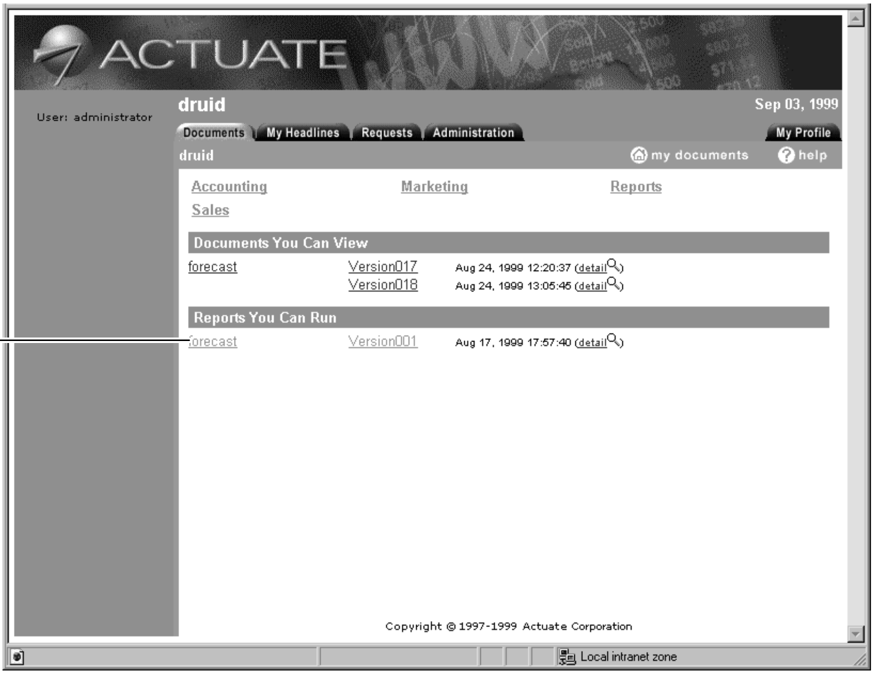
○ Delete objects on 08/30/1999 (mm/dd/yyyy) at 15:58:00 (hh:mm:ss)

□ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels

□ News

Copyright © 1997-1999 Actuate Corporation



Clicking this link generates the ?Request directive shown on the status line.

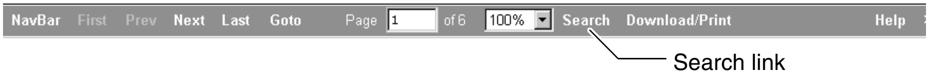
See also Submit

requestSearch

Display the search request page.

Syntax http://<webserver>/acweb/<reportserver>/<folder>/<file>?requestSearch

Description The requestSearch directive allows users to access the search request page to specify criteria for DHTML report searches. Users display the search request page by clicking Search on the navigation bar.



Search link

For information about designing URL's to link to a specific location in a report, see Chapter 10, "Programming for report viewing events," in *Programming e.Reports*.

Example The following requestSearch example displays the search request page for the Marketing plans DHTML report.

```
http://sales/acweb/north/Marketing/plans.roi?requestSearch
```

See also searchReport

searchReport

Specify search criteria for DHTML reports.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
   ?searchReport[&format=<format>][&startingPoint=<start>]
   [&hits=<hits>]&<search list>
```

Pararmeter

format=<format>
Optional. Specifies the search results format. Possible formats are:

- DISPLAY, the standard DHTML display format. This is the default search results format.
- ANALYSIS, available if you have Actuate e.Analysis installed.
- CSV, comma separated values. Download only.
- TSV, tab separated values. Download only.

startingPoint=<start>
Optional. Specifies the first result set data to return, beginning with the first result set. The default value, 0, returns the first result set.

hits=<hits>
Optional. Specifies the total number of results to return.

<search list>
Required. One or more search criteria, optionally followed by select criteria, separated by &'s. The syntax for <search list> is:

```
&<search criterion>[&<select criterion>][&<search criterion>] ...
```

The syntax for <search criterion> is:

```
<class>[.<variable>][:include | :exclude] [=”[value]”]
```

<class>
Required. The fully qualified name of a component class, such as the control component class. If you are creating a searchReport URL to embed in a report or web page, examine the report's .rod file in Actuate e.Report Designer or e.Report Designer Pro to determine the fully qualified name of the component class you wish to search on.

<variable>
Optional. The class variable on which to search. If you do not specify <variable>, the search uses the default value for the class.

<value>
Optional. The value or value expression to search for. If you do not specify <value>, the search uses the empty string, "".

Value expressions can include relational operators, logical operators, and metacharacters. For example, <,>,!, *.

If the value expression contains special characters (>, <, &, ;, @, #) or has one or more spaces at the beginning or end of the expression, enclose the expression in double quotes. For example, C3=">10" specifies a search for a string containing the characters ">10". The value expression C3=">10" does not specify a search for all C3 values greater than 10. ContentField=" 6* " specifies a search for a string containing the characters " 6* ". The value expression ContentField=" 6* " does not specify a search for all ContentField values containing the number 6 followed by any other characters.

:include|:exclude
Optional. The select modifier. The select modifier determines whether or not a specified select criterion, or component instance, is included in the search results. Specify :include to include the component instance in the search result. Specify :exclude to use the search criteria for the component class, but do not display the found instances.

For example, if you build a URL with the select modifier EmpSal:exclude=>0, the search uses the value EmpSal >0 as a select criterion, but does not return values for EmpSal in the search results.

The modifier :include is the default. The modifier is case insensitive.

The syntax for <select criterion> is:

<class>:select[=[<value>]]

<class>
Required. The fully qualified name of a component class. For example, OrderTitleFrame::OrderNumber.

:select
Required. Tells the search to unconditionally include the component in the select results. The value of select overrides the value of the :include | :exclude search modifier.

<value>
Optional. The value of :select. Possible values are:

- TRUE
- FALSE

The values true and false are case insensitive. If you specify :select and <value> is not FALSE, the default value for :select is TRUE. For example, the following select criteria are identical:

```
C1:select
C1:select=true
```

Description The searchReport directive specifies search criteria for DHTML reports. Report developers can embed searchReport directives in Actuate reports or specify them as hyperlinks in any HTML document.

Report developers can build HTML forms that users to use to specify search criteria. This section explains how to build search URL's that return the requested search results.

For more information about performing searches on DHTML reports using the search request page, see "Searching DHTML reports" in Chapter 1, "Using ReportCast."

For information about designing URL's to link to a specific location in a report, see Chapter 10, "Programming for report viewing events," in *Programming e.Reports*.

Working with searchReport

The following guidelines apply to working with the searchReport URL directive.

- The searchReport parameters format, startingPoint and hits are keywords. If a component class has one of these keywords as its name, the search interprets the component class name as a search URL parameter.
- URL parameters are case-insensitive.
- The search uses only the last occurrence of a parameter, search criterion, or select criterion if you repeat them within a search URL.
- Place search parameters before search and select criteria.
- Enclose search values that contain spaces, and special character such as & and *, within double quotes.
- Fully qualified component names that appear in different report sections must have unique names.
- If you specify the :include|:exclude search modifier and do not specify a select criterion, the search uses the search modifier to assemble the select list.

If you do not specify the search modifier, :include is the default.

Example The following searchReport example shows how to build a <select criteria> HTML form.

Building a <select criteria> HTML form

Suppose you want to create an HTML form to execute custom searches in the Customer report. You want to provide the ability to find all customers with a particular credit rank, and you decide to use a checkbox to allow your users to indicate whether to include the credit rank along with the customers' names and phone numbers.

The search criteria are:

```
CreditRank=*&CustomerName=*&PhoneNumber:select
```

Your web server machine name is Greenland, your e.Reporting Server is RServer, and the customer report is in the Customer folder. The HTML form is similar to the following:

```
<FORM NAME="emp_search" METHOD=POST
ACTION="http://Greenland/acweb/RServer/Customer/cust.roi?searchReport">
  <INPUT TYPE="hidden" NAME="format VALUE="DISPLAY">
  <INPUT TYPE="hidden" NAME="startingPoint" VALUE="true">
  <INPUT TYPE="text" NAME="CustomerName:include" VALUE="*">
  <INPUT TYPE="text" NAME="CreditRank:exclude" VALUE="*">
  <INPUT TYPE="checkbox" NAME="CreditRank:select" VALUE="TRUE"
CHECKED>Show Credit Rank
  <INPUT TYPE="hidden" NAME="PhoneNumber:select" VALUE="TRUE">
  <INPUT TYPE="submit" VALUE="Find...">
</FORM>
```

You need to use both the CreditRank:exclude and CreditRank:select because of the way that web browsers process form checkbox data. The previous example's search requires using CreditRank as part of the search criteria. But if the CreditRank:select checkbox is not checked, web browsers ignore CreditRank:select.

For example, here is the query string when CreditRank:select is checked:

```
searchReport&CustomerName:include=*&CreditRank:exclude=*&CreditRank:
select=TRUE
```

CreditRank:select=TRUE overrides CreditRank:exclude, so CreditRank is included in the search results.

Here is the query string when CreditRank:select is not checked:

```
searchReport&CustomerName:include=*&CreditRank:exclude=*
```

So the search proceeds with CreditRank=*, but CreditRank data is not included as part of the search results.

See also requestSearch

Sort and direction

Specify how to display the contents of a Report Encyclopedia folder. Use with the Sort directive.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>
   [?sort=<field>] [& direction = [desc | asc]] [files]
```

Pararmeter

?sort=<field>
The sort directive, followed by the sort field.

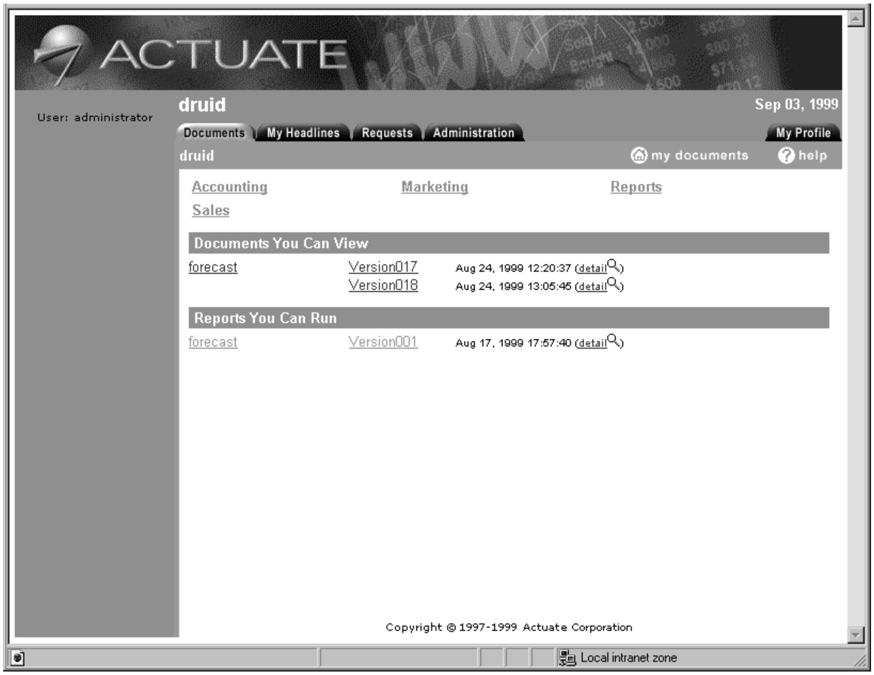
direction = [desc | asc]
The direction of the sort, descending (desc) or ascending (asc). Ascending is the default.

files
Lists only the files that this folder contains. Does not list subfolders.

Description

Use the sort directive to specify the sort order for a list of Report Encyclopedia folder contents. Use the direction directive to specify whether the sort order should be ascending or descending.

When you tell ReportCast to display the contents of a folder on the report server, it displays the folder's files sorted by name and version.



Accounting Marketing Reports
Sales

Documents You Can View

forecast	Version017	Aug 24, 1999 12:20:37 (detail)
	Version018	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast	Version001	Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

You can choose to display a folder's contents in different ways. For example, you can decide to view reports sorted by size, with the smallest reports listed first.

Specify the sort and direction directives in any order. Remember to separate directives and their options with ampersands (&). The ampersand (&) is the standard command separator character for URLs.

The following table lists the valid sort fields.These sort fields are associated with the Name, Type, Owner, Date, and Size links on the HTML page that displays the contents of a folder.

<field> value	Description
Name	Sorts the folder contents by name and version number
Type	Sorts the folder contents by file type
Owner	Sorts the folder contents by owner
Created	Sorts the folder contents by creation date
Date	Sorts the folder contents by modified date
Size	Sorts the folder contents by size

Example In the following example, the URL displays Report Encyclopedia reports sorted by size, with the largest reports listed first.

```
http://sales/acweb/north/Marketing?sort=size&direction=desc
```

See also Files

Status

Display status for the specified report generation request.

Syntax

```
http://<webserver>/acweb/<reportserver>/_ _requests/<request id>?status
```

Description The Status directive displays the current status for the specified report generation request. Status information includes the report name, e.Reporting server running the report, and all status messages that the e.Reporting Server generates as it generates the report.

The Detail directive displays the same information. Detail is available from the Request Status button on the Request Confirmation page.

Example The following Status directive example displays the status for request %3A%202001-05-15%2016%3A01%3A46%201, running on the e.Reporting Server named North.

```
http://sales/acweb/north__requests/
ID%3A%202001-05-15%2016%3A01%3A46%201?status
```

See also Detail

Submit

Submit a report execution request to the e.Reporting Server.

Syntax

```
<reportserver path>/<folder>/<rox> | <rov>
   ?Submit&variable=value[&variable=value...]
```

Pararmeter

<rox> | <rov>
The report executable or parameter values file from which to generate the report.

variable=value
The variable=value pairs define the report parameters and the values to use to generate the report. Values may never include quotes. Use %20 to represent spaces in any string values you define for the report. Use %7B to represent the

opening brace, {, for a date/time expression parameter. Use %7D to represent the closing brace, }, for a date/time expression parameter.

Variables are case-sensitive. You must specify date and time values in the format appropriate for your locale.

The following table lists some standard submit parameter variables. You define report parameters as part of your report design in the e.Report Designer Professional. Report parameters are case-sensitive. Refer to them exactly as you define them in e.Report Designer Professional.

Variable	Description
_ _accessToGrant	The type of access to grant automatically to those roles that have permission to view the report. Permission is granted by a combination of the _ _channels and _ _exclude values. If the output is a secure DHTML report, grant Secure Read access. Otherwise, grant Read access.
_ _ageDays	Optional. Use with _ _ageHours to determine how long output objects exist before they are deleted. Use only if _ _archivePolicy is set to age. _ _ageDays can be any positive number.
_ _ageHours	Optional. Use with _ _ageDays to determine how long output objects exist before they are deleted. Use only if _ _archivePolicy is set to age. _ _ageHours can be any positive number.
_ _archiveBeforeDelete	Optional. Indicate whether or not to archive the output objects of the current request before deleting them, according to _ _archivePolicy's setting. This has no effect if _ _archivePolicy is set to folder. Set to True to archive objects before deleting them. The default value is False.
_ _archivePolicy	Optional. The archive policy to implement for the objects created as output for the current request. Values are folder, age, and date. Set folder to use the archive policy that is already set for the folders to which the output is distributed. Set age to delete objects older than a specific time period. Set date to delete objects on a specific date.

Variable	Description
_ _channels	Optional. Name of a channel to notify of this request. You can notify more than one channel.
_ _dateToDelete	Optional. The date on which to delete the output objects of the current request. Use only if _ _archivePolicy is set to date. _ _dateToDelete must be a date in a locale-specific format. The default format is mm/dd/yyyy.
_ _exclude	Optional. The name of a role that must not have access to the report. You can exclude more than one role.
_ _groups	Optional. The name of the group to notify of this request. You can notify more than one group.
_ _limit	Optional. Indicate whether or not to limit the number of versions of the output files for the current request. Set _ _limit to limit to curtail the number of versions. Any other value means that the number of versions is unlimited.
_ _limitNumber	Optional. The number of versions to which to limit the output files for the current request. Use only if _ _limit is set to limit. _ _limitNumber can be any positive number.
_ _onceDate	Required for once schedules. The date on which to run the report, for reports with _ _scheduleTypes of once. Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is mm/dd/yy. The current date is the default.
_ _onceTime	Required for once schedules. The time at which to run the report, for reports with _ _scheduleTypes of once. Must be in the appropriate format for your locale. For example, the format for the US (enu) locale is hh:mm:ss. The current time is the default.

Variable	Description
_ _outputname	Optional. Name of the output. ROI file for this report. The default is <report>.roi, where <report> is the name of the ROX file. The name can include a date/time expression enclosed in braces, {}, to ensure a unique output name.
_ _overwrite	Optional. New to create a new version of this report. New is the default. Old otherwise.
_ _priority	Optional. The importance that generating this report has in the e.Reporting Server priority. An integer value between 1 and 1000, or Other. 200 is low, 500 is medium, 800 is high. 500 is the default. To set _ _priority, use a radio button group in the HTML requester form.
_ _priorityValue	Optional. Indicates request priority, if the value of: _ _priority is not Other _ _priorityValue is an integer between 1 and 1000
_ _recurringTime	Required for recurring schedules. The time at which to run the report. Set only if reports _ _scheduleTypes is recurring. Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is hh:mm:ss.
_ _schedulePeriod	Required for recurring schedules. How often to run the report, and on which day(s). Choose a day of the week. _ _schedulePeriod values are Every Day, Weekdays, Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Saturdays, Sundays, First Day of the Month, Last Day of the Month. All values are case-sensitive. Every Day or Weekdays. Set only if _ _scheduleType is recurring.

Variable	Description
_ _scheduleType	Required. The type of schedule: immediate, once, or recurring. Immediate is the default.
_ _timeToDelete	Optional. The time on the date specified by _ _dateToDelete on which to delete the output objects of the current request. Use only if _ _archive Policy is set to date. _ _timeToDelete must be a time in a locale-specific format. The default format is hh:mm:ss.
_ _users	Optional. The name of the user to notify of this request. You can notify more than one user.
_ _versionname	Optional. A string value for the new version name of this report. The value can include a date/time expression enclosed in braces, {}, to ensure a unique version name.
_ _wait	Optional. Indicate whether or not to wait for the report to complete, then immediately display the first page. Use only if _ _scheduleType is set to “immediate”. Possible values are wait, nowait, and waitDeleteROV. Choose wait to instruct ReportCast to generate the report then immediately display the first page. Choose nowait to instruct ReportCast to display the request status page while the report generates. Choose WaitDeleteROV to instruct the e.Reporting Server to delete any temporary ROV files after report generation completes. Default is to wait for the report to complete.
BundleRox	True to bundle the data along with the generated ROI file. False otherwise. False is the default value.
Headline	A descriptive tagline for a report. Displays on the Channel Contents page. Use the character string %20 to represent spaces in the headline string.

Description

The Submit directive submits requests for execution in the e.Reporting Server. Users typically submit requests by clicking a Submit button in a requester form, but you can also generate requests and submit the parameters directly in a submit URL.

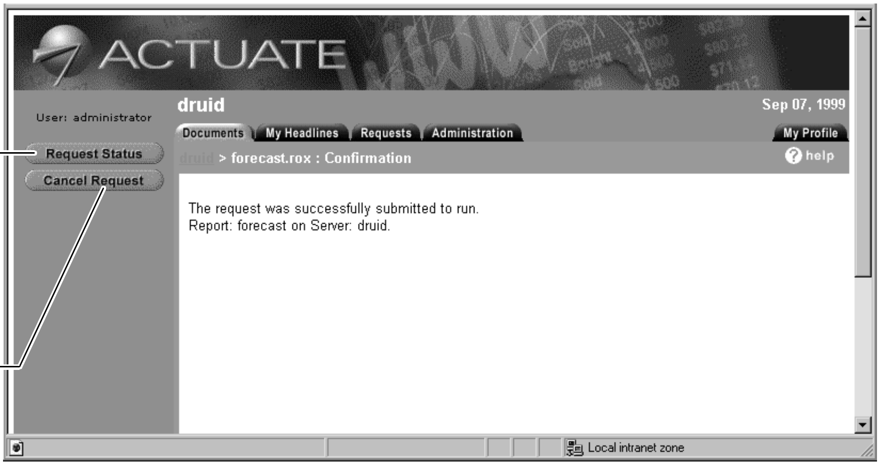
Choosing the Submit Request button at the bottom of the form starts the report generation process. When you choose Submit Request, the form submits the parameter values you specified to ReportCast. ReportCast then issues the Submit directive, which displays a page with information about the report generation request.

The e.Reporting Server places the resulting ROX file in the user's home folder, if the user has a home folder, or in the same folder as the ROI file.

Always include the ?Submit directive when you create custom requester forms. For example, the following directive sets up a submit button:

```
<form method=”POST” action=”$(REPORTURL)?Submit”>
<input type=”image” value=”Submit Request” img src=”MySubmitBtn.gif”>
```

This directive is typically associated with a button or link. The following illustration shows the results of the ?Submit directive.



The request was successfully submitted to run.
Report: forecast on Server: druid.

Clicking this button generates the ?__requests/<request id>?Detail directive, which displays the status of the request.

Clicking this button generates the ?Drop directive that cancels the request.

Example

The following Submit directive example submits the report executable Mosales.rox to the e.Reporting Server, Caligari. Its headline is Monthly Sales Report. It is on a recurring schedule to run on the last day of each month at 1:30 am. Notification goes to the user Admin and groups Sales and Mktg.

```
http://mustique/acweb/caligari/Test/
MoSales.rox?Submit&Headline=Monthly%20Sales%20Report&_ _scheduleTy
pe=recurring&_ _schedulePeriod=Last%20Day%20of%20the%20Month&_ _re
curringTime=01:30:00&__users=Admin&__groups=Sales&__groups=Mktg
```

The following Submit directive example submits the report executable MonthlyReport.rox to the e.Reporting Server, West. The output name includes the current month. The version name includes the current date. The e.Reporting Server deletes any temporary ROV files when the report finishes generating.

The sequence %7B represents the opening brace, {. The sequence %7D represents closing brace, }.

```
http://mustique/acweb/west/Sales/Monthlies/
MonthlyReport.rox?Submit&__outputname=Monthly%20Report%20for%20m
onth%20%7Bmm%7D.roi&__versionname=%20Generated%20%7Bmm-dd-
yyyy%7D&__wait=waitDeleteROV
```

For example, if the current date is March 1, 2001, the output and version name is Monthly Report for month 03.roi Generated 03-01-2001.

See also *Using e.Reports* for more information about specifying a date/time expression as an output name and version name parameter.

submitSearch

Submit a search request and return the search result page.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
    ?submitSearch&startingPoint=<start>&amount=<amount>
```

Pararmeter

startingPoint=<start>
Specifies the first result set data to return.

amount=<amount>
Specifies the number of result hits to return. If amount is not specified, Actuate returns all results beginning with startingPoint.

Description The submitSearch directive submits a search request and returns the search results page. The search criteria are included in the search form data, contained in the URL.

Users generally submit search requests by filling out the search request page, then clicking the Search Now button.

The LimitSearchUI variable controls whether or not users can specify more than one field as search criteria. If LimitSearchUI is set to false, users can specify as many search criteria as they want. If LimitSearchUI is set to true, users can specify only one search criterion. The default is to allow users to specify more than one field as search criteria.

To allow users to specify only one search criterion, edit the Searchrequest.achtml template and set the LimitSearchUI variable to true.

The default Searchrequest.achtml code is as follows:

```
// This variable controls Limit Search UI feature.  This feature limits
// the user to enter search value in the first search criteria field only.
// To enable this feature, set LimitSearchUI to true.
var LimitSearchUI = false;
```

For information about designing URL's to link to a specific location in a report, see Chapter 10, "Programming for report viewing events," in *Programming e.Reports*.

See also extractSearchResults
requestSearch
searchReport

Versions

Display a list of available versions of a file when the file the user is viewing becomes unavailable.

Syntax http://<webserver>/acweb/<reportserver>/<folder>/<file>?Versions

Description The Versions directive displays a list of the available versions of a report. Available reports may expire after specific lengths of time. It is possible that a particular DHTML report could expire and become unavailable as you view it. When a report becomes unavailable, ReportCast displays a list of the available versions of the report. Click the version you want to view. ReportCast displays that version of the report in DHTML format.

Example The following Versions directive displays the list of available versions of the Marketing plans report.

```
http://sales/acweb/north/Marketing/plans.roi?Versions
```

View

Use the LRX to display a report one page at a time in a web browser.

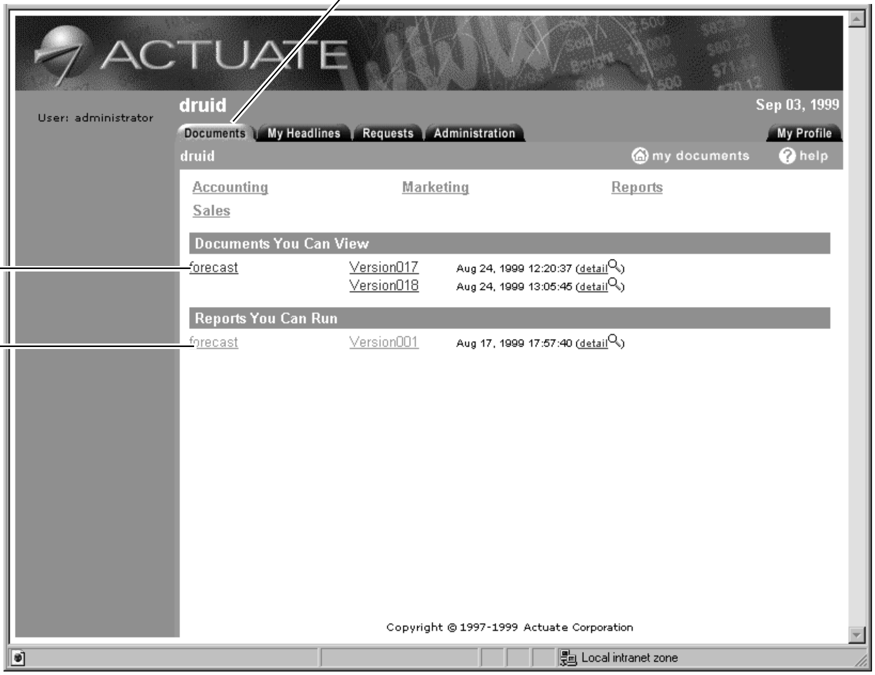
Syntax http:/<webserver>/acweb/<reportserver>/<folder>/<file>?View

Description The View directive runs the Actuate LRX to view a report in a browser window. You can download an entire report document or download only the pages you need. For large reports, the second method, called demand paging, saves time and network resources.

You can use the View directive to view HTML reports (ROW files). Demand paging does not apply to HTML reports. In HTML reports, the View directive displays the entire report in the browser. Use the LRX's navigation buttons to go to particular pages in a report, or move to the report's first, next, previous, or last pages.

The View directive is associated with links in the Name column of the table that displays all the report files in a Report Encyclopedia folder. What happens when you use the link depends upon the type of document. For example, clicking a report document's (ROI's) link allows you to view the report. Clicking a report executable's (ROX's) link allows you to run the report. If you hold your cursor over the icon, the status bar displays information about the link.

Display items in the Reports folder



Documents You Can View

forecast Version017 Aug 24, 1999 12:20:37 (detail)
Version018 Aug 24, 1999 13:05:45 (detail)

Reports You Can Run

forecast Version001 Aug 17, 1999 17:57:40 (detail)

Copyright © 1997-1999 Actuate Corporation

View the report document

Run the report executable

Example The following View directive example displays the Marketing plans report in the LRX.

```
http://sales/acweb/north/Marketing/plans.roi?View
```

See also ViewDefault

ViewDefault

Display a report in a format determined by the user's preferences and viewing privileges.

Syntax http://<webserver>/acweb/<reportserver>/<folder>/<file>?ViewDefault

Description The ViewDefault directive displays the report according to the user's preferences and privileges. If the user's report viewing preference is Dynamic HTML (DHTML), ReportCast displays reports in the web browser in DHTML format. If the user prefers viewing reports using the LRX, but does not have read permission on the report, ReportCast displays the report in DHTML format.

Example The following ViewDefault example displays the Marketing plans report in the user's default viewing format.

```
http://sales/acweb/north/Marketing/plans.roi?ViewDefault
```

See also View

ViewFrameset

Display a report in DHTML format.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
    ?ViewFrameset&page=<page>
```

Pararmeter **page=<page>**
Specifies the number of the initial page to display. The default initial page number is the first page of the report.

Description The ViewFrameset directive generates the toolbar for report navigation and displays the initial frame set for viewing a report in DHTML format. Users generally view DHTML report pages using the navigation bar. The navigation bar includes buttons for next, previous, and first and last pages.

Example The following ViewFrameset directive displays page 3 of the Marketing plans report in DHTML format.

```
http://sales/acweb/north/Marketing/plans.roi?ViewFrameset&page=3
```

See also View
ViewDefault
ViewPage

ViewPage

Retrieve a particular page of a report.

Syntax Save a PDF report for printing or display a DHTML report from the navigation bar:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
   ?ViewPage&format=<format>&action=<action>&page=<page>
   &mode=<mode>&scalingFactor=<scale>&operation=<operation>
```

Set whether to download or view reports in PDF format:

```
http://<webserver>/acweb/<reportserver>/<folder>/
   <file>?ViewPage&format=<format>&action=<action>
```

Display a page from a set of search results or from the report's table of contents:

```
http://<webserver>/acweb/<reportserver>/<folder>/
   <file>?ViewPage&format=<format>&componentID=<ID>
   &scalingFactor=<scale>
```

Pararmeter **format=<format>**
Specifies the format in which to display the report. Format is any format that the view process supports, such as XML structure or page-view formats, or DHTML. The valid values for format are:

- DHTML—a compressed DHTML format that uses Cascading Style Sheets (CSS). This is the default format.
- DHTMLLong—an uncompressed DHTML format. Use this format if your browsers do not support CSS.

action=<action>
Specifies the action to take. Use with the format keyword. Possible values are:

- save—Save the report on the user's local disk in PDF format
- view—View the report in PDF format in Adobe Acrobat Reader

page=<page>
Specifies the number of the initial page to display. The default initial page number is the first page of the report.

mode=<mode>
Indicates the page to display relative to the current page number. Values for mode are:

- First—displays the first page of the report
- Last—displays the last page of the report

- Previous—displays the page that precedes the current page
- Next—displays the page that follows the current page
- Specific—displays the page specified by the page keyword

scalingFactor=<scale>
Specifies the size of the component or the zooming percentage. A scaling factor of 100 indicates the original size of the component. A scaling factor of 50 displays the component at 50% of its original size. 100 is the default scaling factor.

operation=<operation>
Specifies whether to view the report in the browser or save the report in PDF format for printing. Valid values are:

- view—view the report in the browser
- print—save the report in PDF format for printing

Description

The ViewPage directive allows displaying DHTML reports or saving them in PDF format.

Users generally display a specific page in DHTML format by typing the page number in the Page field of the navigation bar.

Users generally download a specific page or range of pages in PDF format for printing by:

1. Clicking the Download/Print link in the navigation bar
2. Specifying the page or pages they wish to print in the Page Range form
3. Clicking the Save PDF button to save the report in PDF format on their local disk

Users print the PDF output after they save it to their local disk. Preparing print output is the default.

Users generally view the PDF version of a specific page or range of pages by:

1. Clicking Download/Print in the navigation bar
2. Specifying the page or pages they wish to print in the Page Range form
3. Clicking View PDF

Users generally display a page from a set of search results by clicking a link in the search results list.

The format keyword specifies the format in which to display the report. Format is any format that the view process supports, such as XML structure or page-view formats, or DHTML.

The action keyword determines whether to save the report on the user's local disk in PDF format, or view the report in PDF format in their web browser. Use the action keyword with the format keyword.

You can use the AC_REPORTCAST_SEND_INLINE_PDF registry key on NT or environment variable on UNIX to set your users' PDF options. Setting AC_REPORTCAST_SEND_INLINE_PDF sets the PDF options for all users accessing a particular ReportCast Server. For more information, see "Configuring PDF viewing and saving options" in Chapter 2, "Administering ReportCast."

Example The following ViewPage directive saves a report in PDF format to the user's local disk.

```
http://west/acweb/sales-rs/Reports/forecast.roi?
ViewPage&format=pdf&action=save
```

See also ViewFrameset

ViewNavigation

Display the navigation bar for viewing a DHTML report.

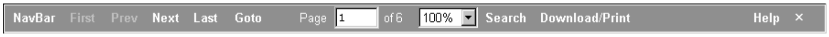
Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
   ?ViewNavigation&page=<page>
```

Pararmeter **page=<page>**
Specifies the DHTML report page to display.

Description The ViewNavigation directive displays the navigation bar, used for viewing DHTML reports. The ViewNavigation directive displays only the navigation bar, not the report. The navigation bar displays when a user clicks a DHTML report document in a report encyclopedia folder.

The following illustration shows the navigation bar.



Example The following ViewNavigation directive displays the navigation bar for the Marketing plans report's page 10.

```
http://sales/acweb/north/Marketing/plans.roi?ViewNavigation&page=10
```

ViewTOC

Retrieve the DHTML report Table of Contents.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
    ?ViewTOC&format=<format>&componentID=<ID>
```

Pararmeter

format=<format>
Specifies the format of the table of contents. XMLDisplay is currently the only supported format

componentID=<ID>
Specifies the component to use as the root of the table of contents. 0 is the default component ID.

Description

The ViewTOC directive displays the DHTML report's table of contents. Users generally display a table of contents for a DHTML report by clicking NavBar in the navigation bar.

ViewDefault

Display a report in a format determined by the user's preferences and viewing privileges.

Syntax

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?ViewDefault
```

Description

The ViewDefault directive displays the report according to the user's preferences and privileges. If the user prefers viewing reports using the LRX, but does not have read permission on the report, ReportCast displays the report in DHTML format.

Example

The following ViewDefault directive displays the Marketing plans report in the user's default viewing format.

```
http://sales/acweb/north/Marketing/plans.roi?ViewDefault
```

See also

View
ViewDefault
ViewPage

Examples of ReportCast directives

The following section provides several examples of directives you enter in the browser's Location or Address box.

The following directive displays the Paradise e.Reporting Server contents. Actserv is the name of the machine on which the web server and ReportCast are installed:

```
http://actserv/acweb/paradise/
```

The following directive displays the contents, sorted by type, of the Reports folder in the Paradise e.Reporting Server:

```
http://actserv/acweb/paradise/reports/?sort=Type
```

The following directive displays only the report files excluding subfolders, sorted by type, in the Reports folder on the Paradise e.Reporting Server:

```
http://actserv/acweb/paradise/reports/?sort=Type&files
```

The following directive displays the request form for running the Detail.rox report:

```
http://actserv/acweb/paradise/reports/detail.rox?Request
```

The following directive submits the request form for running the Detail.rox report:

```
http://actserv/acweb/paradise/reports/detail.rox?Submit
```

The following directive displays all the scheduled factory requests on the Paradise e.Reporting Server:

```
http://actserv/acweb/paradise/_ _scheduled
```

The following directive displays a list of all the channels to which the user is subscribed. This same directive displays the contents of the user's completed folder:

```
http://actserv/acweb/paradise/_ _completed
```

The following directive displays the contents of channel Reports. The contents take the form of a list of the completed requests available for viewing:

```
http://actserv/acweb/paradise/_ _channels/Reports
```

The following directive displays the channel subscription form:

```
http://actserv/acweb/paradise/_ _channels?retrieveDetails
```

The following directive submits the channel subscription form:

```
http://actserv/acweb/paradise/_ _channels?submitDetails
```

Chapter

5

ReportCast scripting language reference

This chapter contains the following topics:

- About the ReportCast scripting language
- ReportCast scripting language reference
- ReportCast scripting language variables

About the ReportCast scripting language

ReportCast scripting language commands tell ReportCast what information to fill in when it generates web pages from the template files. ReportCast templates determine the design and presentation of generated web pages. The templates contain a mixture of normal HTML tags and Actuate extensions. The Actuate extensions are the ReportCast scripting language.

The ReportCast scripting language supports creating custom reporting applications and integrate them with web applications. You can create sets of templates for different audiences and uses, such as:

- Custom templates for specific departments, such as Sales or Accounting
- Internationalized templates for generating web pages in a particular language, such as Japanese, French, Spanish, or English

The elements of the ReportCast scripting language are:

- Commands. Commands tell ReportCast how to insert information when it generates web pages. Commands are case insensitive and must appear in an HTML comment that starts with #Actuate. Each scripting language command must start at the beginning of the line in the template file. Do not place spaces or tabs before the HTML comment containing the scripting language command.
- Variables. ReportCast substitutes variable names with real data when it generates web pages. Variable names are case insensitive and may appear anywhere in a line.

ReportCast scripting language reference

This section presents the ReportCast scripting language commands and their syntax. Each command entry includes a general description of the command and a summary of its options.

Scripting language commands are of the following types:

- Sequential. These are commands that are executed in sequence, from beginning to end, as ReportCast encounters them in a template.
- Iteration. These are commands that loop through lists of objects. Objects include files, folders, properties, and requests. Iteration commands provide some of the same features as e.Report Designer Professional. You can specify the equivalents of Before, After, Content, and IfEmpty sections of reports.
- Selection. These commands provide structures for case statements and conditional statements.

Scripting language commands use the following syntax:

```
<!-- #Actuate <directive> [options] -->
```

Parameter **#Actuate**

- Required.
- Identifies a command.

<directive>

- Required.
- Tells ReportCast what to do.

[options]

- Optional.
- Lists one or more options for the control directive.

The following table lists the scripting language commands.

Command	Description
If	Conditionally includes a particular HTML fragment.
Include	Places the contents of the specified scripting language file into the current file.
List	Iterates through the contents of a list. This command is the most complicated. ReportCast provides many types of lists.
Set	Specifies the value of a scripting language variable.
With	Works with a specified web object.

List command

Loop through the contents of a list.

Syntax

```
list keyword [options]
      ifempty
      before
      content
      after
   end list
```

Description List command provides the ability to loop through a set of lists defined by ReportCast. The ifempty, before, content, and after sections are optional. You can define sections of lists using these optional keywords. Lists begin with list keyword and end with end list. You can define sections of lists using the optional ifempty, before, content, and after keywords. If they are not present, anything between the list and end list keywords is considered to be the content section. Each section begins with its keyword, and ends at the next keyword.

The following table describes each list element.

List element	Description
after	Optional. Specifies content to appear after the main list section. Similar to e.Report Designer Professional's After slot.
before	Optional. Specifies content to appear before the main list section. Similar to e.Report Designer Professional's Before slot.
content	Optional. Specifies content to appear in the main list section.
end list	Required. Specifies the end of the list.
ifempty	Optional. Specifies an action to take if the list is empty.
keyword	Required. Specifies the type of list to process.

List keywords

List keywords specify the type of list to process. The next sections describe the List command's keywords in more detail. The following table describes the List keywords.

Keyword	Description
Channels	Lists the channels available on the report server or the channels to which the current user is subscribed.
Criteria	Lists the search criteria that were submitted as part of a search request.
Expirationpolicies	Lists the expiration policies for the current file.

Keyword	Description
Files	Lists the files in the current Report Encyclopedia folder.
Folders	Lists the folders in the current Report Encyclopedia folder.
Grants	Lists all members that have explicitly been granted permissions for the current item.
Items	Lists the folders and files in the current Report Encyclopedia folder.
Members	Lists all members on the current Report Encyclopedia. Members include all users, roles, and groups.
Notifications	Lists the users and groups to notify when the current request completes.
ParamGroups	Lists all parameter groups for a request.
Parameters	Lists all parameters for a request or group.
Pathlink	Lists the path links that apply to the current folder or file.
PrinterProps	Lists the properties of the printer being used for the current request.
Privileges	Lists the privileges each user and role has for the current item.
ReportServers	Lists the report servers to which ReportCast has open connections.
RequestForms	Lists the available custom and standard request pages that run the report associated with the current ROI, ROW, ROX, or ROV file.
Requests	Lists all requests.
Resultfields	Lists the fields that make up the search results.
Resulthits	Lists the search result hits returned by the view process.
Schedules	Lists the string descriptions of schedules for the current request.
Searchformats	Lists the supported formats for presenting the results of search operations.
Status	Lists the status items for a request.
Tocitems	Lists the table of contents items to display.
Values	Lists the values for a submitted request.

Keyword	Description
Versions	Lists the versions, if any, for the current file.
Viewpageformats	Lists the supported formats for displaying DHTML report pages.

List channels command

Lists channels to which the current user is subscribed.

Syntax list [all] channels

Description Provides a list of channels to which the user is subscribed. For example, the standard template for the Personal Channel, Personalchcontents.achtml, displays the list of channels to which the user is subscribed in the left sidebar. Use the All option to get a list of all available channels on the current e.Reporting Server.

List channels is available from all templates except for Admin.achtml.

Example The following example lists all channels available on the current e.Reporting Server:

```
<!-- #Actuate list all channels -->
```

List criteria command

Syntax list criteria

Description Lists the search criteria for a search query. The list criteria command is available from the search results page, named Searchresults.achtml.

List expirationpolicies command

Syntax list expirationpolicies

Description Creates a list of the object aging expiration policies that apply to the current Report Encyclopedia file. The expirationpolicies command is available from the file detail page, Filedetail.achtml.

List files command

Lists the objects in the current Report Encyclopedia folder.

Syntax list files [(type [, type...])] [sort by field [desc]]

Description List files lists Report Encyclopedia objects in the current encyclopedia folder. You can list all files or just files of a particular type. You can specify the sort order for the files or use the sort order specified in the URL directive that generates the file list web page. This command lists only the files in the current folder. It does not list the contents of other folders within the current folder.

Example The following example shows how to list the files in a Report Encyclopedia folder:

```
<!-- Start the list -->
<!-- #Actuate list files -->
<!-- Define the list heading -->
<!-- #Actuate before -->
<TABLE>
<TR>
<TH>Name</TH> <TH>Owner</TH> <TH>Date</TH>
<!-- Here is the main list content -->
<!-- #Actuate content -->
<TR>
<TD> $(FileName) </TD>
<TD>$(OwnerName)</TD>
<TD>$(Created)</TD>
</TR>
<!-- Now, define the end of the list -->
</TABLE>
<!-- Here’s what to do if the list is empty -->
<!-- #Actuate ifempty -->
This folder does not contain any report items.
<P>
<!-- #Actuate end list -->
```

List files type clause

Specify the kinds of files to list. File types are indicated by the extension of the file in the Report Encyclopedia, for example, .row, .roi, .rox. Types are case insensitive. You can specify wildcard characters in the type clause.

Wildcard	Matches
?	Any one character
*	Any number of characters
...	All files not previously matched by a preceding pattern

Example

The following example lists all the report executables in the current folder:

```
<-- #Actuate list files (ROX)
```

List files sort by clause

Specifies how to sort the files in the list.

Description

Takes the sort order from the URL directive, by default. The default sort direction is ascending. You can override the default sort order by specifying the sort by clause in your list files command. Use the desc keyword to change the direction of the sort to descending. You can sort by only one field at a time. ReportCast cannot sort by multiple fields.

The following table lists the available sort fields.

Sort field	Description
Date	Sorts by file creation date.
Name	Sorts by file name and version number.
Owner	Sorts by file owner name.
Size	Sorts by file size.
Type	Sorts by file type.

Example

The following example lists all report executables, sorted by creation date, in ascending order:

```
<!-- #Actuate list files (ROX) sort by date
```

List folders command

List all the folders in the current Report Encyclopedia folder.

Syntax list folders [sort by field [desc]]

Description List folders lists all folders in the current Report Encyclopedia folder. You can sort the list by specific fields. You can also specify whether the sort should be ascending or descending. The default sort order is ascending.

The following example lists all folders in the current Report Encyclopedia folder, sorted by owner, in descending order:

```
<--! #Actuate list folders sort by owner desc -->
```

List folders is available from the folder listing page, named Folderlist.achtml.

See also List files sort by clause

List grants command

Lists all members that are explicitly granted permissions on the current item, either a Report Encyclopedia object or a channel.

Syntax list grants [all] {(users | roles | groups [,...])}

Description List grants lists those object privileges that are explicitly granted to the current user or role. It does not include those privileges that the current user or role inherited from a role to which they belong. The set of users returned by list grants does not include the set of users who inherit all their privileges from their roles.

List grants is available from the following templates:

- File detail page, named Filedetail.achtml
- Channel detail page, named Channeldetail.achtml
- Request page, named Request.achtml

List grants can be nested inside the list versions or list items commands or in a template for a folder, encyclopedia item, or channel. This command supports allowing or disallowing the current member to execute a report, depending upon whether the user has privileges on an item.

Use the All option to list all members, not only those that have privileges on the current folder, file, or channel. If the current user does not have administrator privileges, the list of privileges returned applies only to the current user and the roles to which that user belongs.

For more information about members, see “List members command,” earlier in this chapter.

Example The following example lists the roles that are granted privileges on the current item. The example is based on the Request.achtml template:

```
    <!-- #Actuate list grants (roles) -->
    <!-- #Actuate if MemberName <> “Administrator” and MemberName <> “All”
    then -->
    <!-- #Actuate Set RoleFound = “True” -->
    <!-- #Actuate end if -->

<!-- #Actuate end list -->
```

List items command

Lists the contents of the current Report Encyclopedia folder.

Syntax list items [sort by field [desc]]

Example The following example lists all the items in the current Report Encyclopedia folder, using the default sort order. The default sort order is ascending:

```
<--! #Actuate list items -->
```

List items is available from the folder listing page, named Folderlist.achtml.

See also List files sort by clause.

List members command

Lists every member in a Report Encyclopedia.

Syntax list members { (users | roles | groups [....])}

Description The term member refers to any user, role, or group in a Report Encyclopedia. You can use list members to generate a pull-down list of users to notify when a particular request completes. The default is to list all members or specify to include only certain types of members, for example, list only users, or roles, or groups. The list is sorted by name.

List members is available from all templates except for the admin page, called Admin.achtml. The list members command cannot be nested inside another list members command or inside the list grants or list notifications commands.

Example The following example iterates through a list of the roles that are members on the current e.Reporting Server and sets up an input checkbox. This example is based on the Channeldetail.achtml template:

```
<!-- #Actuate list members (roles) -->
   <!-- #Actuate if MemberName <> “Administrator” and MemberName <>
“All” then -->
       <input type=”checkbox” name=”_ _Roles” value=”$(MemberName)”>
       $(MemberName)
   <!-- #Actuate end if --> <!-- MemberName -->
<!-- #Actuate end list --> <!-- members -->
```

List notifications command

Creates a list of users and groups to notify when the current request is complete. The list is sorted by name.

Syntax list notifications

Description Supports user notification when a request to run a report completes. The standard requester form, Request.achtml, displays the list of available channels. To notify a particular channel when a request completes, users click the appropriate check box. You can also modify the Status.achtml template to show the list of channels to be notified when the user checks the status of the request.

The list notifications command is available from the following templates:

- The confirmation page, named Confirm.achtml
- The cancellation page, named Cancel.achtml
- The request details page, named Requestdetail.achtml
- The status page, named Status.achtml

List pathlinks command

Syntax list pathlinks

Description Creates a list of the path links that apply to the current folder or file. The list is empty if there is no current folder or file. Use this command to find the individual elements of a folder or file’s complete path name. For example, list pathlinks allows cascading style sheet properties to control the colors of Report Encyclopedia objects’ links on the reporting web page.

List pathlinks is available from the following templates:

- The folder list page, called Folderlist.achtml
- The file details page, called Filedetail.achtml

- The confirmation page, called Confirm.achtml
- The request details page, called Requestdetail.achtml
- The status page, called Status.achtml
- The cancellation page, called Cancel.achtml

Example The following example is a fragment from Folderlist.achtml. It displays the folder path name with each directory name separated by a small folder icon:

```
<!-- #Actuate list pathlinks -->
<!-- #Actuate ifempty -->
<tr>
   <td valign="bottom" nowrap>
   <table border="0" cellpadding="0" cellspacing="0">
       <tr>
           <td><img src="$(imgFolderBig)" align="top" border="0"></td>
           <td><font size="3" face="Arial"><strong>$(ServerAlias)</strong>
</font></td>
       </tr>
   </table>
   </td>
</tr>
<!-- #Actuate before -->
<tr>
   <td valign="bottom" nowrap>
   <table border="0" cellpadding="0" cellspacing="0">
       <tr>
           <td><img src="$(imgFolderBig)" align="top" border="0"></td>
           <td><font size="3" face="Arial"><strong>
<!-- #Actuate content -->
<!-- #Actuate if Hyperlink <> "" then -->
                   <a href="$(Hyperlink)">$(DisplayName)</a>&nbsp;
<img src="$(imgFolder)" border=0>&nbsp;
<!-- #Actuate else -->
                   $(DisplayName)
<!-- #Actuate end if -->
<!-- #Actuate after -->
           </strong></font></td>
       </tr>
   </table>
   </td>
</tr>
<!-- #Actuate end list -->
```

List paramgroups, list parameters commands

Use these commands to create custom requester forms.

Syntax

```
list paramgroups

list parameters
```

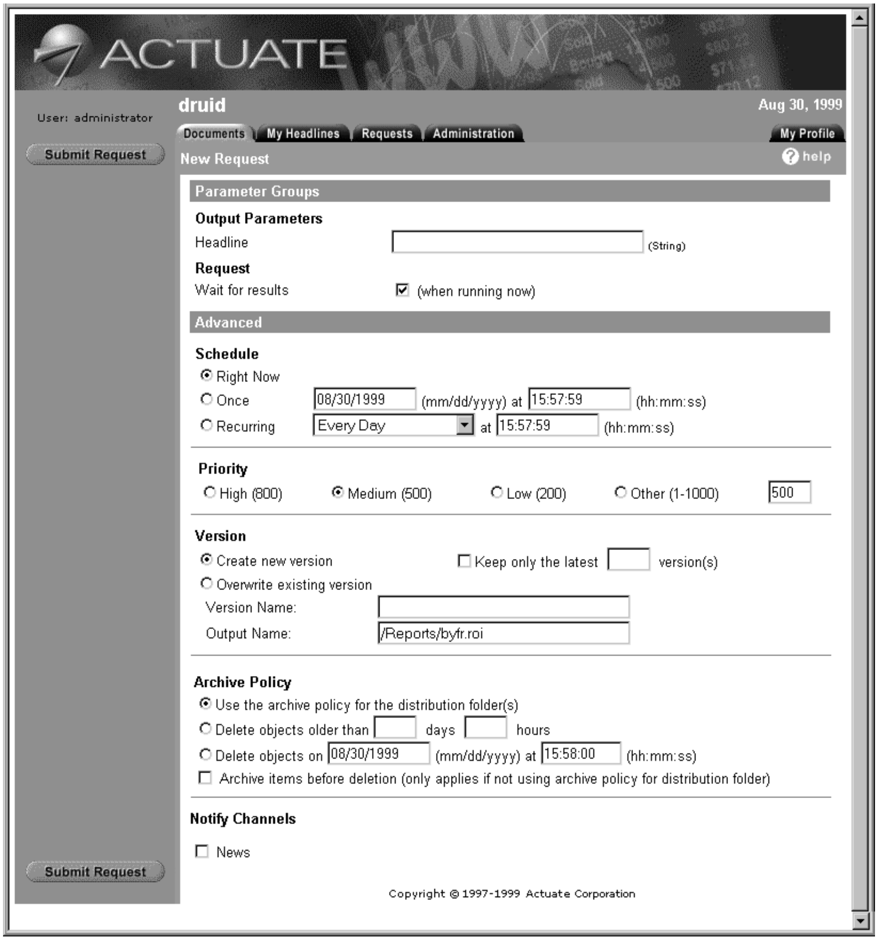
Description

List paramgroups and list parameters provide a requester template to generate a custom requester form.

When users send a request to run a particular report, ReportCast generates a requester form. The requester form lists the available report parameters and provides users the opportunity to fill in the parameters they want. For example, users can run a standard sales report, using parameters such as region, sales office, quarter, and so on.

If many users from particular regions or departments plan to run the standard report regularly, you can provide custom requester forms that provide default values for parameters the users would otherwise have to insert. For example, you can create a requester form for the Western Region sales group, another form for the Eastern Region sales group, and still other requester forms for the European, Asian, and South American sales groups.

The following illustration shows a standard requester form for a report. The form lists all the parameters required for the report, including channels to notify when the report finishes running. The parameters are listed by group.



Parameter Groups

Output Parameters

Headline ▭ (String)

Request

Wait for results ☑ (when running now)

Advanced

Schedule

● Right Now

○ Once 08/30/1999 (mm/dd/yyyy) at 15:57:59 (hh:mm:ss)

○ Recurring Every Day at 15:57:59 (hh:mm:ss)

Priority

○ High (800) ● Medium (500) ○ Low (200) ○ Other (1-1000) 500

Version

● Create new version ☐ Keep only the latest ▭ version(s)

○ Overwrite existing version

Version Name: ▭

Output Name: /Reports/byfr.roi

Archive Policy

● Use the archive policy for the distribution folder(s)

○ Delete objects older than ▭ days ▭ hours

○ Delete objects on 08/30/1999 (mm/dd/yyyy) at 15:58:00 (hh:mm:ss)

☐ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels

☐ News

Copyright © 1997-1999 Actuate Corporation

The requester template, Request.achtml, can list parameters by group or by name.

Use list parameters to list the parameters alphabetically. Use list paramgroups to list the parameters by group. List parameters can be nested. List paramgroups cannot be nested.

List parmgroups and list parameters are available from the following templates:

- Standard requester page, named Request.achtml
- Request confirmation page, named Confirm.achtml
- Request details page, named Requestdetails.achtml
- Request status page, named Status.achtml
- Cancelled request confirmation page, named Cancel.achtml

List printerprops command

Lists property names and values for the printer being used for the current request.

Syntax list printerprops

Description Provides information about the printer being used to print out the current request. List printerprops is available from the request details page, named Requestdetail.achtml, if the request was either Print or Execute and Print.

List privileges command

Lists all the privileges granted to the current user or role for the current encyclopedia item or channel.

Syntax list privileges

Description Lists all the privileges available in the current Report Encyclopedia. Use list privileges within a list grants command to list each of the privileges granted to the current user or role for the current Report Encyclopedia item or channel. If you use list privileges in the context of a Report Encyclopedia item, the returned list does not include privileges inherited from a role to which the current user or role belongs.

List privileges is available from the following templates:

- The channel details page, named Channeldetail.achtml
- The file details page, named Filedetail.achtml

List reportservers command

Lists the report servers to which ReportCast has open connections.

Syntax list reportservers

Description ReportCast opens a connection to a particular e.Reporting Server each time a user accesses an e.Reporting Server through ReportCast. Users with administrative privileges can check the open connections from ReportCast to a particular e.Reporting Server from the ReportCast administration page. Open connections use system resources. If response time is slower than desirable, you can close open connections from the administration page.

List reportservers is available from the admin page, named Admin.achtml.

List requestforms command

Lists the available requester forms.

Syntax list requestforms

Description Lists the requester forms available to run the report specified by the URL directive or associated with the file name in the URL directive. List requestforms is available only from ReportCast's Requester Menu page.

List requestforms is available from the requester menu page, named Requestermenu.achtml.

List requests

Lists the requests on the current report server or for the current user.

Syntax list requests

list {active | scheduled | completed} requests [sort by {name | date | status}]

Description ReportCast uses the list requests command to generate a list of requests when it receives a URL directive for the _ _active, _ _completed, or _ _scheduled request folders. List requests is available only from ReportCast's Request Folder page.

The first form lists all requests on the current e.Reporting Server.

The second form lists requests for the current user. Use this form to create a web page other than a request folder page, for example, to display the user's completed requests on a Report Encyclopedia's main page, so their completed reports are immediately available.

List requests is available from the request list page, named Requestlist.achtml.

List resultfields command

Syntax list resultfields

Description Lists the fields that are part of the search query's results.

You can call list resultfields within the scope of a particular search result. You can then access the value and component ID for each field in that result.

You can call list resultfields outside the scope of any particular search result. You can them access only the name of each field contained in the search query's results.

The list resultfields command is available from search results page, named Searchresults.achtml.

List resulthits command

Syntax list resulthits

Description Lists the search result hits returned by the search query. The list resulthits command is available from the search results page, named Searchresults.achtml.

List schedules command

Lists the schedule associated with the current request.

Syntax list schedules

Description Supports specifying simple schedules for running reports, such as daily, right now, or once at a specified time and date. The e.Reporting Server's Administrator Desktop also allows you to specify complex schedules, such as twice weekly plus once a month on a specific day. You can view complex schedules using the list schedules command, even though you cannot specify those commands using ReportCast.

The list schedules command is available from the request details page, named Requestdetail.achtml.

List searchformats

Syntax list searchformats

Description Lists the supported formats for presenting the results of search operations. Use the $(FormatValue) variable to iterate through the list of available formats.

You can use the values searchformats values to create or modify the ?searchReport URL.

The list of available formats depends upon the configuration of Actuate e.Reporting Server. The number of installed products, add-ins, and custom converters determines the formats that are available. For example, the ANALYSIS format is available if the e.Analysis option is installed.

The following are the supported search formats.

- XMLDisplay, a display format
- Download formats:
  - CSV, comma-seperated values
  - TSV, tab-seperated values
  - ANALYSIS, e.Analysis data

The list searchformats scripting language command is available to the following templates:

- The search request page, named Searchresults.achtml
- The search results page, named Searchresults.achtml
- The navigation bar, named Viewnav.achtml
- The report table of contents, named Viewtoc.achtml

Example The following example lists the available search formats.

```
<P>Here are the available search formats:<P>
<!-- #Actuate list searchformats -->
<!-- #Actuate before -->
<!-- #Actuate content -->
$(FormatValue)<br>
<!-- #Actuate after -->
<!-- #Actuate end list -->
<P>
```

List status command

Generates a request status page.

Syntax list status

Description The request status page lists the status items available for the current request. List status is available from the following templates:

- Confirmation page, named Confirm.achtml
- Request status page, named Status.achtml
- Cancelled request confirmation page, named Cancel.achtml
- Request details page, named Requestdetail.achtml

Example The following example iterates through a list of status items for reports on a particular e.Reporting Server and displays them on a status page. This example is taken from the Status.achtml template:

```
The following is the status of running $(ReportName) on $(Server):
<!-- #Actuate list status -->
<!-- #Actuate before -->
<center>
<table border=0 bgcolor=black>
<thead valign=top>
<tr bgcolor=”#804080”>
<th>Time<th>Status
<tbody>
<!-- #Actuate content -->
<tr bgcolor=”E0E0E0”><td>$(DateAndTime)<td>$(StatusText)</tr>
<!-- #Actuate after -->
</table>
</center>
<!-- #Actuate end list -->
```

List tocitems command

Syntax list tocitems

Description Lists the items in the DHTML report Table of Contents. The list tocitems command is available from the view table of contents page, named Viewtoc.achtml.

List values command

Generates a request confirmation page.

Syntax list values [setonly]

Description Lists the values that the user set for request parameters. For more information about request parameters, see “List paramgroups, list parameters commands,” earlier in this chapter.

The list values setonly option lists only those parameters for which the user entered a value. You can use this with the list paramgroups command to group the values by parameter groups.

List values is available from the following templates:

- Request confirmation page, named Confirm.achtml
- Request details page, named Requestdetail.achtml
- Request status page, named Status.achtml
- Cancelled request confirmation page, named Cancel.achtml

List values can be nested inside the list paramgroups command.

Example The following example iterates through a values list and displays the values on a request confirmation page. This example is taken from the Confirm.achtml template:

```
<!-- #Actuate list values -->
<!-- #Actuate content -->
<table>

<!-- Display the values heading -->

        <tr>
<td>&nbsp;</td>
<td><font size=”2” face=”Arial”>Description</font></td>
<td>&nbsp;</td>

<!-- If the value is null, display a space -->
<!-- #Actuate if Value = ““ then -->
<td><font size=”2” face=”Arial”>&nbsp;</font></td>

<!-- Otherwise, display the value -->
<!-- #Actuate else -->
<td><font size=”2” face=”Arial”>$(Value)</font></td>

<!-- #Actuate end if -->
</tr>
</table>

<!-- #Actuate end list -->
```

List versions command

List the available versions of the current file.

Syntax list versions

Description Each time the report runs, e.Reporting Server creates a new version of the report file unless you specify otherwise. If a user attempts to view a version of a file in Dynamic HTML (DHTML) that is no longer available, list versions lists the available versions of that file.

List versions is available from:

- The file versions page, named Fileversions.achtml
- The file details page, named Filedetail.achtml

Example

```
<!-- #Actuate list versions
```

List viewpageformats

Syntax list viewpageformats

Description Lists the available formats for displaying DHTML report pages. Use the $(FormatValue) variable to iterate through the list of available formats.

You can use the viewpageformats values to create or modify the ?ViewPage or ?GetReportData URL dynamically in ReportCast templates.

The list of available formats depends upon the configuration of Actuate e.Reporting Server. The number of installed products, add-ins, and custom converters determines the formats that are available.

The following are the supported display formats.

- DHTML
- DHTMLLong
- DHTMLRaw
- PDF
- XMLCompressedDisplay
- XMLDisplay

The list viewpageformats scripting language command is available to the following pages:

- The search request page, named Searchresults.achtml
- The search results page, named Searchresults.achtml
- The navigation bar, named Viewnav.achtml
- The report table of contents, named Viewtoc.achtml

Example The following example lists the available search formats.

```
<P>Here are the available search formats:<P>
<!-- #Actuate list searchformats -->
<!-- #Actuate before -->
<!-- #Actuate content -->
```

```
$(FormatValue)<br>
<!-- #Actuate after -->
<!-- #Actuate end list -->
```

If command

Conditionally includes HTML in the generated web page.

Syntax

```
if expr then
   elseif expr then
   else
   end if
```

expr
An expression.

Expressions have the following formats:

```
expr and expr
expr or expr
(expr)
not expr
var = “string”
var <> “string”
var = var
var <> var
```

Description Supports including HTML in generated web pages based on specific conditions. For example, use the If command to provide a particular format for particular types of files or hide files the current user is not authorized to see.

Include command

Includes another template file with the current template file.

Syntax

```
include filename
```

Description Including other template files with the current template file provides reusability for your HTML template code. The included template file can include other template files, which may include other template files, and so on. Included template files must be both readable and located in the current search path’s directory tree. In other words, they must be local to the report server. You cannot include templates that reside on other file systems or directory structures.

The Include command is useful for including a standard element in several templates. It simplifies maintenance to make changes to one file, rather than to several files. For example, include the standard company logo in the Logo.achtml template file and standard company copyright information and links to help in the Copyright.achtml template.

Example The following example includes a template in the current template file:

```
<!-- #Actuate include “stdfooter.achtml” -->
```

With command

Makes the specified object’s variables and list statements available.

Syntax

```
with objectname
   statements
   end with
```

Description Includes a named object’s variables and list statements in the current object’s scope. Objects have the following values.

Object value	Description
InputObject	Makes available the Folder Item for the ROX file of the request.
OutputObject	Makes available the Folder Item for the ROI or ROW of the request.
ReportDefinition	Makes available the Report Definition for the current request, which provides the parameter, parameter group lists, and other properties.
ValuesObject	Makes available the Folder Item for the ROV file of the request.

Example The following example makes the ReportDefinition object available, then iterates through its parameter groups list:

```
<!-- #Actuate with ReportDefinition -->
<!-- #Actuate list paramgroups -->
<!-- #Actuate content -->
<!-- (Actions taken on parameter groups here) -->
<!-- #Actuate end list -->
<!-- #Actuate end with -->
```

Set command

Assigns a value to a variable.

Syntax

set var=”string”

set var1=var2

var, var1, var2
Unique variable names. Variable names are not case-sensitive.

string
The value of the character string to assign to var.

Description

Assigned values can be character strings or other variables’ values. Once a variable is assigned, it is available within the current template using the following syntax:

$(VariableName).

Example

The following example checks whether or not a folder contains items, and if so, sets the variable FolderHasItems to True. The following example is taken from the Folderlist.achtml template:

```
<!-- #Actuate list items -->
<!-- #Actuate content -->
<!-- #Actuate if FolderPath<>"/" or ( ItemBaseName<>"$$$TempROVs" and
ItemBaseName<>"$$$CUTCOPY" ) then -->
<!-- #Actuate Set FolderHasItems = "True" -->
<!-- #Actuate end if -->
<!-- #Actuate end list -->
```

ReportCast scripting language variables

The ReportCast scripting language variables are placeholders for data within the template files. ReportCast substitutes actual data values for the variables when it generates the web page. The variable substitution process enables creating HTML pages dynamically, depending on the directive sent. Dynamic HTML pages support page generation that reflects the constant data changes in the database.

ReportCast variables provide current, run-time access to information about a wide variety of topics including the ReportCast services, e.Reporting Server, the available folders and files, schedules, ReportCast channels, requests, privileges, and parameters.

Variables use the following syntax:

```
$(variable)
```

All variable names are case-insensitive. For example, the following command tells ReportCast to generate a page that displays the name of a report and its owner's name:

```
The name of the report is $(ReportName) and the owner is $(OwnerName).
```

If the report name is SalesEastQ401.roi, and the owner is SalesMgr, the generated page displays:

```
The name of the report is SalesEastQ401 and the owner is SalesMgr.
```

ReportCast variables

ReportCast variables return information about the machine running ReportCast, as well as information about ReportCast itself. ReportCast variables are available on all templates. The following tables list the scripting language variables.

Variable	Purpose
AdminURL	Displays the URL to ReportCast's administration page.
AgentVersion	Displays the product name and version of ReportCast.
ElapsedTime	Displays the elapsed time since ReportCast was started.
Host	Displays the name of the web server machine that is connected to the ReportCast server. This is the web server machine itself, not a proxy server. To resolve Host to a host name, DNS on the ReportCast Server machine must be able to find a host name for the address of the web server machine. Otherwise, this value is blank.
Now	Displays the current time on the machine running ReportCast. Time format is localized.
NumberOfCommands	Displays the number of commands that ReportCast has processed since the last time it was started.
ReportServerName	Displays the report server name that the user specified during login. ReportServerName is available only from the login form and only if ReportCast is not set up with Basic Authentication.

Variable	Purpose
StartDateAndTime	Displays the date and time that ReportCast was started.
TemplateLoadDateAndTime	Displays the date and time that ReportCast's template file cache was last cleared.
Today	Displays the current date on the machine running ReportCast. Date format is localized.

Web server variable

The following table lists the ReportCast variable that returns information about the web server on which ReportCast is running. This variable is available on all templates.

Variable	Purpose
WebServerVersion	Displays the product name and version of the web server in which ReportCast is running.

Web request variables

The following table lists the ReportCast variables that return information about the current command that ReportCast is processing. These variables are available on all templates.

Variable	Purpose
CommandName	Displays the name of the command appended to the URL.
ObjectURL	Displays the URL only for the object that is the target of the URL.
PreviousURL	Displays the URL of the web page previous to the current page.
RemoteAddress	Displays the dotted-decimal IP address of the browser.
RemoteHost	Displays the fully resolved host name of the browser.
TemplateName	Displays the name of the template without the directory path.
URL	Displays the full URL sent to ReportCast.

Variable	Purpose
UserName	Displays the name of the current user.
UserAgent	Displays the contents of the User Agent, for example, Mozilla/3.01 WinNT.

e.Reporting Server variables

The following table lists ReportCast variables that return information about the e.Reporting Server named in the current request's URL. These variables are available in all templates except the ReportCast administration page.

Variable	Purpose
ActiveRequestsURL	Displays the URL for the list of active requests on the e.Reporting Server.
ChannelsURL	Displays the URL to the listing of the current user's channels.
CompletedRequests URL	Displays the URL for the list of completed requests on the e.Reporting Server.
HomeFolderURL	Displays the URL for the listing of the current user's home folder on the e.Reporting Server. If no home folder is defined for the current user, HomeFolderURL has the same value as RootFolderURL.
IsAdministrator	True if the current user belongs to the administrator group of the current e.Reporting Server. False otherwise.
RootFolderURL	Displays the URL to the listing of the root folder of the e.Reporting Server.
ScheduledRequests URL	Displays the URL for the list of scheduled requests on the e.Reporting Server.
Server	Displays the name of the e.Reporting Server.
ServerAlias	Displays the descriptive string name of the e.Reporting Server. If there is no descriptive name, ServerAlias is the e.Reporting Server's machine name.
ServerFeatureLevel	Indicates the e.Reporting Server level. 0 is None, 1 is Standard, and 2 is Advanced.
ServerVersion	Displays the full product name and version of the e.Reporting Server.

Variable	Purpose
ViewFormatDefault	Displays the system-wide default set by the e.Reporting Server administrator, either DHTML or ROI.
ViewFormatPreference	Contains the current user’s format preference for viewing reports. DHTML, ROI, or SYSTEM DEFAULT. SYSTEM DEFAULT uses the viewing preference set by the e.Reporting Server administrator.

Note that the value for ServerAlias comes from the ACSERVERALIAS property of the root folder on the e.Reporting Server. Use the Navigator to set the e.Reporting Server’s name property, which sets the ACSERVERALIAS value.

View session variables

The following table lists the ReportCast variables that return information about the current view session. The view session variables are available whenever the user views a DHTML report.

Variable	Purpose
CurrentPage	The number of the DHTML report page currently being viewed.
CurrentTOCParentID	The component ID of the current Table of Contents entry or node.
FormatValue	The value of a supported search or viewing format.
TotalPages	The total number of pages in the DHTML report.

Table of contents item variables

The following table lists the ReportCast variables that return information about the current table of contents item. The table of contents item variables are available whenever the user makes use of a DHTML table of contents.

Variable	Purpose
ComponentID	The current table of contents item's component ID.
Expandable	True if the table of contents item has subitems that appear when a user expands the item by clicking on the item's + sign, False otherwise.
Text	The displayable name of the current table of contents item.

Search criterion variables

The following table lists the ReportCast variables that return information about the current search criterion. The search criterion variables are available whenever the user performs a search on a DHTML report.

Variable	Purpose
ClassName	The name of the class to which the current search criterion belongs.
ComponentID	The component ID of the control that the current search criterion applies to.
ControlName	The name of the control that the current search criterion applies to. For example, Office name.
SearchValue	The expression that the control's value must match. For example, Boston or >1000. SearchValue is blank if the user did not enter a search expression for the control, but selected it as part of the search results.
Selected	True if the user selected the control as part of the search results, False otherwise.

Search result set variables

The following table lists the ReportCast variables that return information about the current search result set. The search result set variables are available whenever the user examines the results returned by a search on a DHTML report.

Variable	Purpose
EndHitNumber	The index of the last hit displayed on the search results page. The range is 1–TotalNumberOfHits.
Format	The value of the format URL.
Frameset	The value of the frameset URL.
NumberOfHitsOnPage	The number of hits displayed on the search results page.
StartHitNumber	The index of the first hit displayed on the search results page. The range is 1–TotalNumberOfHits.
TotalNumberOfHits	The total number of result hits returned by the search.

Search result field variables

The following table lists the ReportCast variables that return information about the current search result field. The search result field variables are available whenever the user examines a specific search result field.

Variable	Purpose
ComponentID	The current search result hit's component ID.
Value	The displayable value of the current search result field.

Current folder variables

The following table lists the ReportCast variables that return information about the current folder. The current folder variables are available whenever the user displays a page that describes a folder, for example, the folder listing page Folderlist.achtml.

Variable	Purpose
FolderHeader	Displays the description of the folder's contents.
FolderName	Displays the name of the current folder.

Variable	Purpose
FolderPath	Displays the full path name of the current folder.
FolderTitle	Displays the title to be used in the HTML <TITLE> tag.
LastModified	Displays the date and time that the folder was last modified.
LinkedFolderPath	Displays a series of lines that provide a hierarchical path to the current folder.
ParentFolderPath	Displays the full path name of the folder that contains the current folder.
ParentURL	Displays the URL of the parent folder. This variable’s value is empty (““) if there is no parent folder.

Folder item variables

The following table lists the ReportCast variables that return information about folder items. Folder items are both folders and the files inside a folder. These variables are available for each item in a list returned by the list items, list files, and list folders commands, and from the File detail page.

Variable	Purpose
AbbreviatedSize	Displays the item size, abbreviated according to the localized display format. For example, 12.5MB or923KB.
AbbreviatedType	Displays the file extension, for example, .rox or .roi.
Comment	Displays a lengthy description of the file, stored as a property in the Report Encyclopedia. Displays on detail pages in the Navigator. Comments can be of any length.
Created	Displays the item creation date and time.
FileSize	Displays the item size.
FileType	Displays the item’s file type.
FileVersion	Displays the item version number.
FileVersionName	Displays the descriptive version name of the file item. If there is no descriptive name, ReportCast displays Version <n>, <n> is the version number.

Variable	Purpose
IsBundledOutput	True if the object is an ROW or a third-party report consisting of HTML pages, False otherwise. View bundled files using the ?View or ?ViewDefault directives.
IsExecutable	True for ROX, ROV, and third-party report executable files, False otherwise. You can request these files using the ?Request directive.
ItemBaseName	Displays the item's base name without the file name extension.
ItemName	Displays the name of the current item.
ItemPathName	Displays the full path name of the item.
LastModified	Displays the date and time that the item was last modified.
LastModifiedBy	Displays the name of the user who last modified the item.
LatestURL	The URL to the most recent version of the file.
LinkedFolderPath	A series of lines that provide a hierarchical path to the current folder item's properties.
NewItem	True if the current item is the first version of a file to be listed in the list items or list files commands, False otherwise. The first version is not necessarily Version 1. It is simply the first item in this particular list. Its position depends on the sorting options used.
NumVersions	Displays the number of versions of the file that correspond to the current item.
ObjectID	Displays the item's unique object ID.
OwnerName	Displays the name of the item's owner.
PathName	Displays the full path name of the item on the e.Reporting Server.
URL	Displays the URL of the item.

Current request folder

The following table lists the ReportCast variables that return information about the request folder. These variables are available from the Request folder page or the Channel contents page.

Variable	Purpose
Entries	Displays the number of requests in the request folder.
FolderName	Displays the name of the request folder: Scheduled, Active, or Completed.

Report definition

The following table lists the ReportCast variables that return information about a report. These variables are available from the Standard Requester page and the Request confirmation page.

Variable	Purpose
CancelURL	Displays the URL for cancelling the current request. Available only from the Request Confirmation page.
DefaultDate	Provides a default date value (the current date) for the report requester form in locale-specific format.
DefaultTime	Provides a default time value (the current time) for the report requester form in locale-specific format.
DownloadURL	Displays the URL for downloading the output of the current request. Available only from the Request Confirmation page.
LastModified	Displays the date and time the report executable was last modified.
ReportName	Displays the name of the report executable, without the file extension.
ReportOutputPath	Displays the default location of the output file for the report (path and file name).
ReportPath	Displays the complete name of the report executable.
ReportURL	Displays the URL of the report executable.

Variable	Purpose
RequestURL	Displays the base URL containing the ID of the current request, without the query portion. RequestURL can be combined with queries in other templates to produce URLs other than the request's URL.
StatusURL	Displays the URL for the status page of the current request. Available only from the Request Confirmation page.
ViewURL	Displays the URL for viewing the output of the request. Available only from the Request Confirmation page.

Request item variables

The following table lists the ReportCast variables that return information about items in a list requests list. These variables are available from the Request status and Request detail pages.

Variable	Purpose
AbbreviatedCurrent Size	Displays the current size of the output of the request, in KB, abbreviated.
AbbreviatedSize	Displays the size of the completed request's output file, in KB, abbreviated.
CancelURL	Displays the URL of the command to cancel the request.
CompletedTime	Displays the date and time at which the current request completed.
CurrentPages	Displays the current number of pages of the request's output.
CurrentSize	Displays the current size of the output of the request, in bytes.
DeleteURL	Displays the URL for the command to delete status information for the current request.
DetailURL	Displays the URL for the request's detail page.
ElapsedTime	Displays the time expired since the start of the request.
Headline	Displays the request headline. This variable is valid only for successfully completed requests.

Variable	Purpose
NextStartTime	Displays the time that the request is scheduled to start.
OutputLocation	Displays the name of the folder containing the request’s output file.
OutputName	Displays the base name of the report document that will be generated from the current request.
OutputPathName	Displays the complete name of the output file, including the full path name and file extension.
OutputVersion	Displays the version number of the completed request’s output file.
OutputVersionName	Displays the version name of the output file of the completed request.
Pages	Displays the number of pages in the completed request’s output file.
PrintRange	Displays the range of pages to be printed for the scheduled request.
Priority	Displays the numeric priority of the request.
ProcessGroup	Displays the name of the process group running the request.
ReportName	Displays the name of the requested report executable (.rox).
ReportPathName	Displays the complete name of the requested report’s executable, including the full path name and file extension.
ReportURL	Displays the URL of the report executable.
ReportVersion	Displays the version number of the requested report executable.
ReportVersionName	Displays the version name of the requested report executable.
RequestedBy	Displays the name of the person submitting the request.
RequestID	Displays the unique ID of the current request.
RequestStatus	Displays the status of the current request: Scheduled, Active, Completed, or Failed.
RequestType	Displays the type of the current report-generation request: Print or Execute.

Variable	Purpose
RequestURL	Displays the base URL for the ID of the current request, without the query portion.
Size	Displays the size of the completed request's output file.
StartTime	Displays the date and time at which the current request started to run.
StatusURL	Displays the URL of the status page for the current request.
ValuesFileName	Displays the name of the scheduled request's .rov file.
ViewURL	Displays the URL for viewing the output of the request. This variable is valid only for successfully completed requests.

Member variables

The following table lists the ReportCast variables that return information about members including users, groups, and roles. Member lists include the list of all members on an e.Reporting Server, the list of members that have privileges for a folder item or channel, and the list of members to be notified about a request.

Variable	Purpose
MemberName	Displays the name of the current user, group, or role.
MemberType	Displays the member's type: User, Group, or Role.

Privilege variables

The following table lists the ReportCast variables that return information about privileges for a current folder item or channel and a current member. These variables are available from the list privileges command.

Variable	Purpose
Granted	Displays True or False depending on the current privilege, folder item or channel, and member.
PrivilegeName	Displays the name of the current privilege.

Request status variables

The following table lists the ReportCast variables that return information about the current request's status. These variables are available from the List status command.

Variable	Purpose
DateAndTime	Displays the date and time of the status element.
StatusText	Displays the text of the status element.

Parameter group variable

The following table lists the ReportCast variable that returns information about related groups of report parameters. These variables are available from the List paramgroups command.

Variable	Purpose
GroupName	Displays the parameter group's name.

Parameter variables

The following table lists the ReportCast variables that return information about report parameters. These variables are part of report definitions and parameter groups. These variables are available from the list parameters command. Use the List parameters command either from the top level of a report definition or within a List paramgroups command.

Variable	Purpose
AdHoc	Displays True if this parameter is ad hoc. Displays False otherwise.
Alias	Displays the parameter's alias, if one exists, or "" if there is no alias.
Default	Displays the parameter's default value.
Description	Displays a description of the parameter's name. This description is the same as Alias if Alias is not empty.
Hidden	Displays True if this parameter is hidden. Displays False otherwise.

Variable	Purpose
HideText	Displays True if this parameter's value must be masked. Displays False otherwise. Useful for masking values such as passwords in the authentication dialog.
Name	Displays the parameter's name. ReportCast passes this name to the factory.
Required	Displays True if this parameter is required. Displays False otherwise.
Type	Displays the type of information that the parameter should contain, for example, String, Number.
Value	Displays the value that the user entered for the current parameter.

Schedule variable

The following table lists the ReportCast variables that return information about the current request's schedule. These variables are available from the List schedules command.

Variable	Purpose
ScheduleDescription	Displays the text description of the request's schedule.

Printer variables

The following table lists the ReportCast variables that return information about the printer being used for the current request. These variables are available from the list Printerprops command.

Variable	Purpose
Name	Displays the name of the current printer property.
Value	Displays the value of the current printer property.

ReportCast Channel variables

The following table lists the ReportCast variables that return information about the current channel. These variables are available from the list channels command or from the Channel detail or Channel contents pages.

Variable	Purpose
ChannelName	Displays the name of the current channel.
ChannelURL	Displays the URL of the current channel.
Comment	Displays a description of the channel, with HTML tags. The description may be of any length.
Expiration	Displays the number of days an item remains on the current channel before it is removed.
LargeIconURL	Displays the URL of the large version of the custom icon for the current channel. This value is empty (“”) if there is no custom icon.
NextPollTime	The time at which the browser must refresh the contents of a channel. This value is an HTTP date, and can be used in a <meta> HTML tag to cause a page to expire from the browser’s persistent cache. Current information about HTTP dates is available from the HTTP specification available from the World Wide Web Consortium’s web site, http://www.w3.org/.
PlainComment	Displays a description of the channel, without the HTML tags. Use PlainComment in multiline input tags on an HTML form.
PollingInterval	Displays the number of minutes between browser refreshes of the current channel’s contents. This number is formatted appropriately for the current configuration. For example, commas separate every three digits in English: 1,000.
RawPolling Interval	Displays the number of seconds between browser refreshes of the current channel’s contents. This number is not formatted.
SmallIconURL	Displays the URL of the small version of the custom icon for the current channel. This value is empty (“”) if there is no custom icon.
Subscribed	Displays True if the current user is subscribed to the current channel. Displays False otherwise.

Error variables

The following table lists the ReportCast variables that return information about error conditions that occur on Report Encyclopedia requests. Error variables are available from the Error page.

Variable	Purpose
ErrorClass	Displays the type of error that occurred: 1 e.Reporting Server Connection Error 2 e.Reporting Server Message Error 3 ReportCast Internal Error 4 ReportCast Parser Error
ErrorNumber	Displays the numeric code for the error. This numeric code is an HTTP status code for all but ReportCast internal errors and parser errors.
ErrorText	Displays a description of the specific error condition, if possible. The description is localized.
RawErrorText	Displays a description of the specific error condition. The text is not localized.
Server	Displays the name of the report server on which the error occurred.

Chapter

6

Internal ReportCast security

This chapter contains the following topics:

- About ReportCast security
- Providing security for ReportCast administration
- Logging on and authenticating users
- Security and the Apache web server
- Customizing the Login dialog
- Setting the idle user session timeout
- Modifying the HTTP unauthorized status code
- About the ReportCast Security Extension
- ReportCast Agent Security Extension library functions
- ReportCast Server Security Extension library functions
- Development considerations
- Enabling RCASE and RCSE

About ReportCast security

A reporting web site is accessible to any user who has a web browser and the URL to the site. How do you ensure that users access only those Report Encyclopedia objects for which they have permission? How do you protect sensitive reports or ensure that only authorized users can perform administrative tasks for ReportCast? This chapter discusses the ReportCast security features and how to use them to provide secure access to your web reporting site.

The types of security you can provide for ReportCast are:

- Administrative security. Assign an administrator login and password to provide security for performing ReportCast's administrative tasks. This security applies to tasks specific to ReportCast only. They are not related to e.Reporting Server administration.
- Default user authentication. Use the default ReportCast and e.Reporting Server facilities to ensure that users access only those reports and other Report Encyclopedia objects for which they have permission.
- User authentication using ReportCast Security Extension (RCSE) or ReportCast Agent Security Extension (RCASE). Use the RCSE or RCASE to customize and control the user login and authentication process.

You can also customize the Login dialog box and specify the unauthorized status code that ReportCast returns when authentication fails.

Actuate fully supports HTTPS, Secure Socket Layers (SSL). The default port for HTTPS is 443. For information about configuring the port numbers, see "Configuring ReportCast port numbers" in Chapter 2, "Administering ReportCast."

Setting the ReportCast security variables

Enabling the ReportCast Agent Security Extension and ReportCast Server Security Extension and setting up secure facilities involves using registry keys on Windows NT or environment variables on UNIX to set values for the RCSE security variables.

Setting security variables on Windows NT

Create new registry keys using Regedit. Be sure to back up your system before running Regedit:

1. Choose Start➛Run.
2. In Run, type:

```
regedit.exe
```

3 Choose OK.

4 In Registry Editor, navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

5 Choose Edit➛New➛String Value.

   A new value item appears on the right side of the registry.

6 Type the name of the security variable.

   For example, if you are setting the registry key for the AC_REPORTCAST_ADMINISTRATOR_USER_NAME security variable, type:

   ```
   AC_REPORTCAST_ADMINISTRATOR_USER_NAME
   ```

7 Set the security variable:

   1 Select the security variable name.

   2 Right-click the security variable name. From the context menu, choose Modify.

   3 In Edit String, type the value for the security variable in the Value data: field. Choose OK.

Setting security variables on UNIX

Set the security environment variables in the user environment in which ReportCast runs. In other words, you must be logged on as the user who runs ReportCast. That is usually the same user who runs the web server, unless ReportCast runs separately from the web server as on HP-UX systems.

The easiest way to set security variables is to set them and export them in the ReportCast startup script, startreportcast.sh. The ReportCast installation places the startreportcast.sh script in the <webserver-home>/actuate//bin directory, where <webserver-home> is the root of your web server installation. Add the setenv, and export statements to the startup script just before the script invokes the ReportCast server, reportcastsrvr.

For example, to set and export the AC_REPORTCAST_ADMINISTRATOR_USER_NAME security variable:

```
AC_REPORTCAST_ADMINISTRATOR_USER_NAME=<user>
export AC_REPORTCAST_ADMINISTRATOR_USER_NAME
```

<user> is the administrator user name.

Providing security for ReportCast administration

You perform administrative tasks for ReportCast from the ReportCast administration web page:

```
http://<webserver>/acweb/__admin
```

<webserver> is the name of the web server that is running ReportCast.

The ReportCast administration web page accesses administrative information for ReportCast. That page cannot be used to administer the Report Encyclopedia.

To enable secure access to ReportCast's administration page, use these registry keys for NT or these environment variables for UNIX:

- AC_REPORTCAST_ADMINISTRATOR_USER_NAME sets the administrator's user name.
- AC_REPORTCAST_ADMINISTRATOR_PASSWORD sets the administrator's password.

You can set a user name and password or leave both blank. For example, a ReportCast installation that is used for testing purposes only or that is accessible only internally may not require security. Setting these values for ReportCast does not affect access to your Report Encyclopedias. For information about setting these security variables, see "Setting the ReportCast security variables," earlier in this chapter.

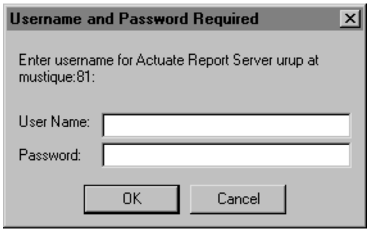
Logging on and authenticating users

Users can log on using their Report Encyclopedia user names and passwords. Another option is using anonymous login. If the anonymous login is set up in the server or Report Encyclopedia, ReportCast automatically logs on users as anonymous.

Logging on as anonymous

The anonymous user allows users to access reports on a particular e.Reporting Server without having to go through a login screen or any sort of authentication. e.Reporting Server administrators set up the anonymous user in the Administrator Desktop, configure it with View Properties privileges on all but the most sensitive objects, and do not assign a password.

When a user attempts to access a secure object, ReportCast displays the Login dialog.



Username and Password Required

Enter username for Actuate Report Server urup at mustique:81:

The user must type a valid user name and password to be able to access the secure object. ReportCast reconnects the user to the e.Reporting Server with the new authentication information and privileges.

Most browsers put already-viewed pages into their cache. That makes viewing web pages faster, but the pages could be out of date. If you reconnect to the e.Reporting Server as a different user, make sure that the web reporting pages you view are current. If they are not, perform a forced reload of the page:

- In Netscape Navigator, hold down the Shift key while choosing Reload.
- In Microsoft Internet Explorer, clear the cache and then reload the page.

For more information about setting up the anonymous user, see Chapter 2, “Managing Report Encyclopedia security,” in *Administering the Report Encyclopedia*.

Logging on as an e.Reporting Server user

If the e.Reporting Server administrator did not set up an anonymous user, Login appears when the user first accesses the e.Reporting Server. When the user enters a valid user name and password, the initial e.Reporting Server view appears in the browser.

Authenticating users using forms

To log in as a different user using a dialog, users must first log out, then shut down their web browser, restart their web browser, and log back in. If you enable form-based login authentication, users can log out and log back in as a different user without shutting down their web browser.

Users with administrator privileges can customize the login and logout forms. Customize the login form by modifying Login.achtml. Customize the logout form by modifying Logout.achtml.

Form-based authentication is the default for Apache web servers. For other web servers, enable form-based login authentication by setting the AC_REPORTCAST_USE_LOGIN_FORM registry key on NT or environment variable on UNIX to True.

Security and the Apache web server

The Apache web server default security process is slightly different from other Unix web servers' default security processes.

Users log in using a form instead of a dialog. For more information, see "Authenticating users using forms," earlier in this chapter.

Customizing the Login dialog

You can customize the Login dialog to a certain extent. The Login text is:

Enter username for <custom text> at <webserver>:

- <custom text> is the text that you can customize.
- <webserver> is the name of the web server to which the user wants to connect. You can include the string $(Server) in your custom text to display the name of the server to which the user is logging in.

Customize Login using the security variable AC_REPORTCAST_AUTHENTICATION_PROMPT. For information about setting this security variable value, see "Setting the ReportCast security variables," earlier in this chapter.

For example, setting the following value for AC_REPORTCAST_AUTHENTICATION_PROMPT:

```
logging in to MyCorp's $(Server) server
```

results in the following Login prompt to the Sales web server from the ERegion server:

```
Enter username for logging in to MyCorp's Sales server at ERegion:
```

Setting the idle user session timeout

If a user is idle for more than 1800 seconds (30 minutes), he must log in again. Users with administrator privileges can change the idle time value by setting the environment variable AC_REPORTCAST_LOGIN_TIMEOUT. Specify, in seconds, the length of time user sessions can be idle. For example, to set the time-out to 15 minutes, add the following code to the startup script just before the script invokes the ReportCast server, reportcastsrvr:

```
AC_REPORTCAST_LOGIN_TIMEOUT=900
export AC_REPORTCAST_LOGIN_TIMEOUT
```

Modifying the HTTP unauthorized status code

When user authentication fails, ReportCast sends the HTTP unauthorized status code, 401, to the browser. You can modify the HTTP unauthorized status code that ReportCast sends to the browser by setting the AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE variable.

Authentication fails if the ReportCast Agent Security Extension or the ReportCast Server Security Extension returns a non-zero value from an authentication function or if login to the e.Reporting Server fails. By default, ReportCast sends back the status code of 401 if authentication fails. The 401 error causes the browser to prompt the user for user name and password. This security variable allows the administrator to prevent clients from trying to authenticate by returning a value other than 401. For information about setting this security variable, see “Setting the ReportCast security variables,” earlier in this chapter.

If you want to customize the text that appears if authentication fails, you can customize the Error.achtml template. For more information about customizing the ReportCast templates, see “About ReportCast templates,” in Chapter 3, “Working with ReportCast templates.”

About the ReportCast Security Extension

The Actuate ReportCast Security Extension controls the way ReportCast uses information associated with requests from the web browser to log on to the e.Reporting Server.

The ReportCast Security Extension consists of the following:

- ReportCast Agent Security Extension (RCASE). RCASE resides with the ReportCast Agent as part of the web server. You can use RCASE to filter and change HTTP headers, including user authentication information.
- ReportCast Server Security Extension (RCSE). RCSE resides with the ReportCast Server. RCSE receives the HTTP headers forwarded from RCASE. RCSE can authenticate the user information with a security database or send the user information to the e.Reporting Server for authentication.

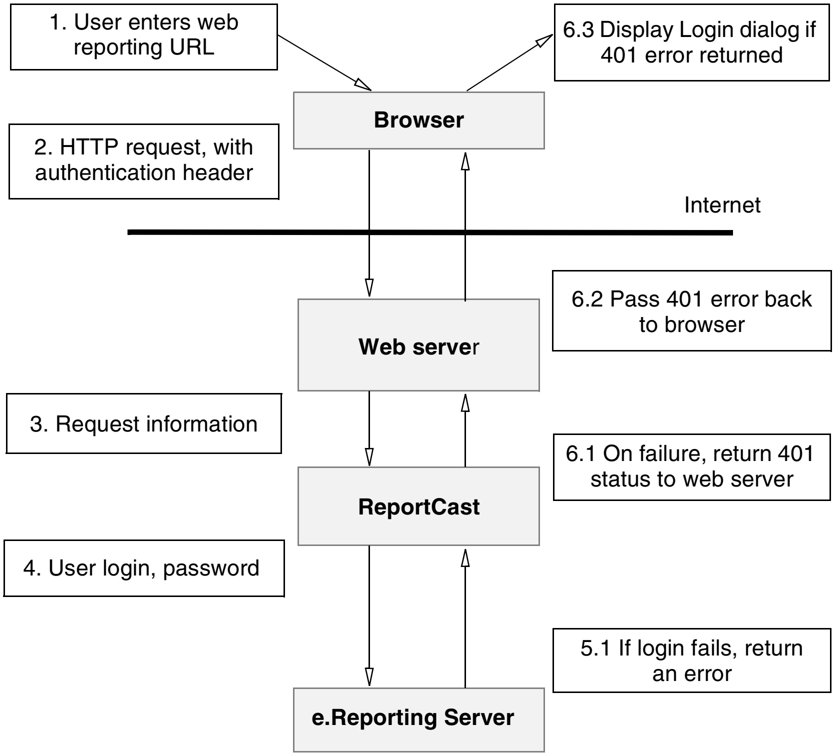
Overview

The Actuate ReportCast Security Extension replaces the default user authentication in ReportCast with external routines. Using the security extension supports developers customizing the interaction among the web browser, ReportCast Agent, ReportCast Server, and e.Reporting Server to control the user login and authentication process.

Authentication means determining whether users are who they say they are. Authorization means determining whether or not users have access rights to a resource.

You usually integrate RCASE and RCSE with the e.Reporting Server Security Extension (RSSE). For more information about RSSE, see *Integrating Actuate e.Reporting Server*.

The following diagram illustrates the default authentication process.


1. User enters web reporting URL
6.3 Display Login dialog if 401 error returned
Browser
2. HTTP request, with authentication header
Internet
6.2 Pass 401 error back to browser
Web server
3. Request information
6.1 On failure, return 401 status to web server
ReportCast
4. User login, password
5.1 If login fails, return an error
e.Reporting Server

The following steps describe the default authentication process:

1 The user enters the web reporting home page URL in the browser.

2 The browser passes the HTTP request to the web server, including the authentication header.

3 The web server recognizes the acweb keyword and passes the request information to ReportCast.

4. ReportCast maps the Basic Authentication header to the user login and password, then passes the user login and password to the e.Reporting Server. If there is no Basic Authentication header, the user login becomes anonymous and the password is null (““). The anonymous login works only if it is already set up on the e.Reporting Server.
5. The e.Reporting Server processes the user login and password:
    1. If login fails, the e.Reporting Server returns an error to ReportCast.
    2. If login succeeds, the e.Reporting Server returns successful to ReportCast.
6. If login failed:
    1. ReportCast returns the HTTP 401 status to the web server.
    2. The web server returns the HTTP 401 status to the browser.
    3. The browser displays Login, and the process begins again.

This authentication process is effective most of the time. It might require the user to log on multiple times, however, if additional security is in place on the reporting web site. This authentication process does not support the Microsoft Internet Information Server Windows NT Challenge/Response security feature. It supports only Basic Authentication.

Using ReportCast Agent Security Extension and ReportCast Server Security Extension, you can override the default authentication process and customize the user name and password that ReportCast uses to access the e.Reporting Server. If you enable RCASE, ReportCast Agent calls it to authenticate users each time it receives a request to connect to the e.Reporting Server. If you enable RCSE, ReportCast Server calls it to authenticate users each time it receives a request to connect to the e.Reporting Server.

RCASE can access HTTP headers and the HTTP body. You can customize RCASE to retrieve the HTTP header and body information, perform authentication against a security database, update the HTTP header and body information, and pass the authenticated request to the ReportCast Server.

RCSE can access the HTTP headers, verify authentication information against a security database, modify the HTTP headers, and pass the request to the e.Reporting Server.

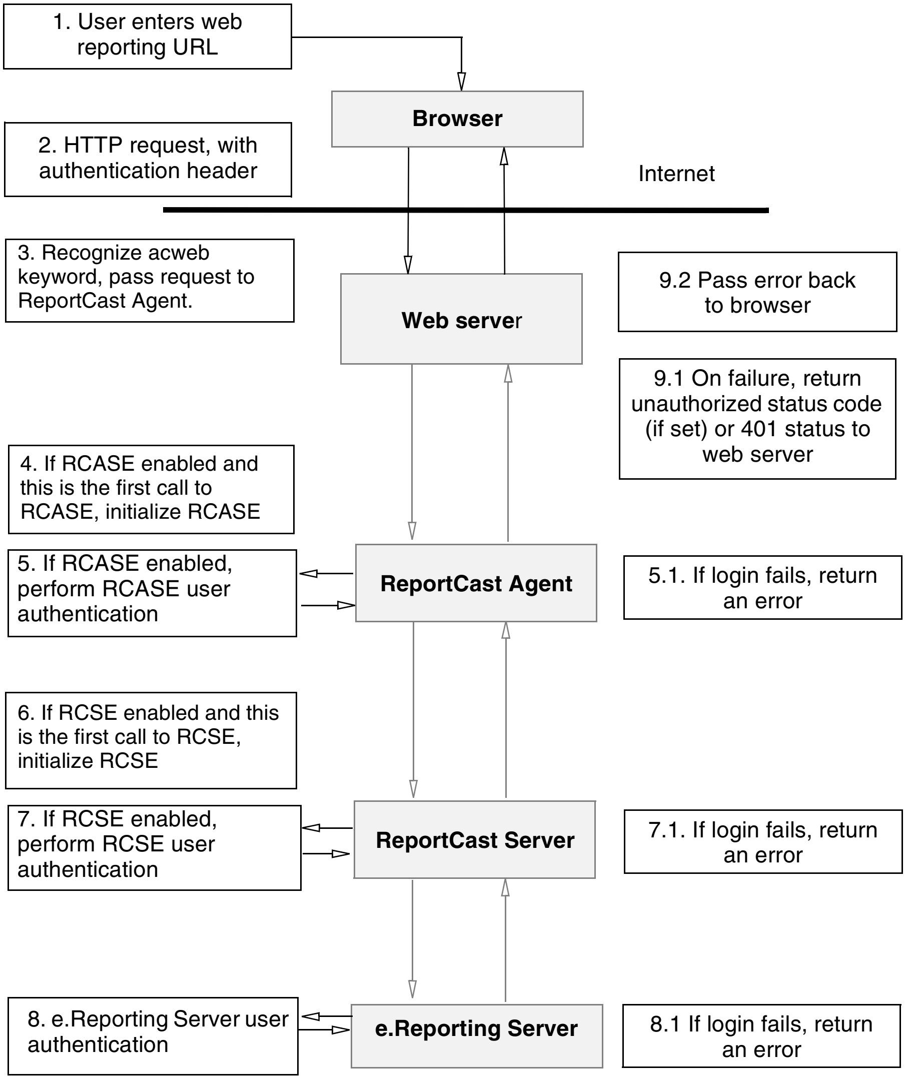
Depending upon your site’s needs, you can customize authentication for the ReportCast Agent, the ReportCast Server, or both. For example, your ReportCast configuration can have one ReportCast Server communicating with several ReportCast Agents, each with its own security requirements. You can:

- Create custom security DLLs for each ReportCast Agent.

- Create a custom security DLL for the ReportCast Server that handles custom security requirements depending upon the ReportCast Agent that sent a request.
- Handle authentication only on the ReportCast Agent.
- Handle authentication only on the ReportCast Server.

You enable the ReportCast Security Extension by creating registry entries in NT or by setting environment variables on UNIX. You must create the registry entries or set the environment variables for each ReportCast Agent Security Extension or ReportCast Server Security Extension. The registry entries or environment variables must reside on the same machine with the ReportCast Agent Security Extension or ReportCast Server Security Extension to which they apply.

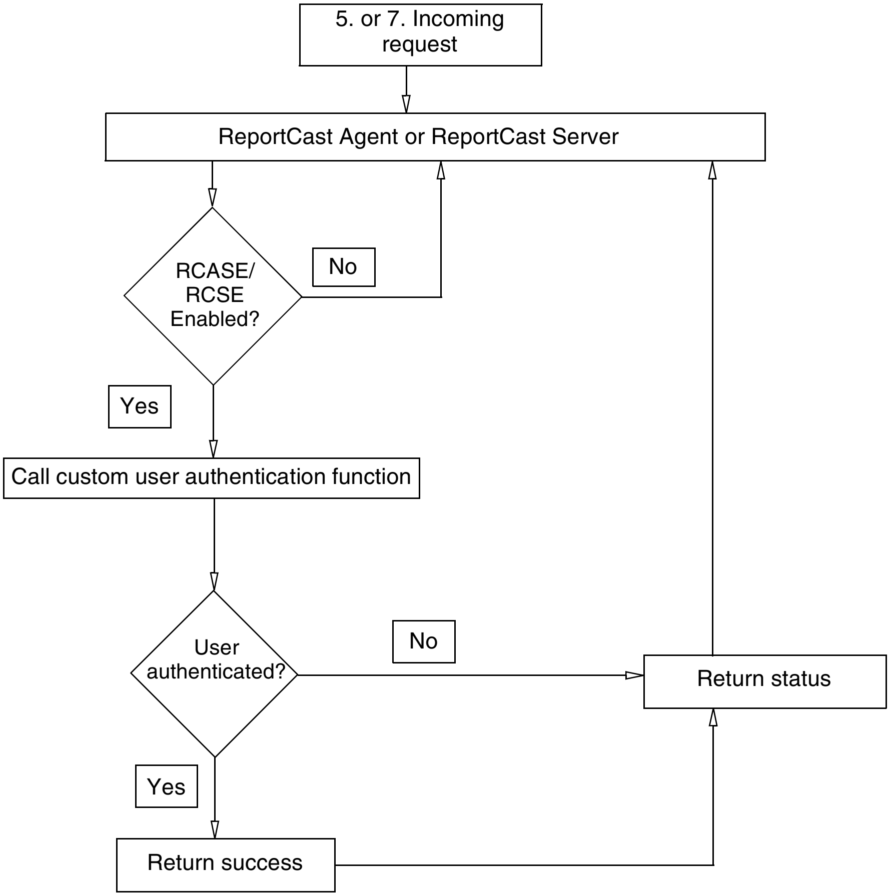
The following illustration shows an overview of the authentication process using the ReportCast Agent and ReportCast Server Security Extensions.


1. User enters web reporting URL
Browser
2. HTTP request, with authentication header
Internet
3. Recognize acweb keyword, pass request to ReportCast Agent.
Web server
9.2 Pass error back to browser
9.1 On failure, return unauthorized status code (if set) or 401 status to web server
4. If RCASE enabled and this is the first call to RCASE, initialize RCASE
5. If RCASE enabled, perform RCASE user authentication
ReportCast Agent
5.1. If login fails, return an error
6. If RCSE enabled and this is the first call to RCSE, initialize RCSE
7. If RCSE enabled, perform RCSE user authentication
ReportCast Server
7.1. If login fails, return an error
8. e.Reporting Server user authentication
e.Reporting Server
8.1 If login fails, return an error

This ReportCast Agent and ReportCast Server Security Extension authentication process is similar to the default authentication process:

1. The user enters the web reporting home page URL in the browser.
2. The browser passes the HTTP request to the web server, including the authentication header.
3. The web server recognizes the acweb keyword and passes the request information to ReportCast Agent.
4. If the ReportCast Agent Security Extension is enabled, and this is the first call to RCASE, ReportCast initializes RCASE.
5. Authenticate the user using your custom security library on UNIX or DLL on NT.
    1. If authentication fails, return an error to ReportCast Agent.
    2. If authentication succeeds, pass the request to ReportCast Server.
6. If the ReportCast Server Security Extension is enabled, and this is the first call to RCSE, ReportCast initializes RCSE.
7. Authenticate the user using your custom security library on UNIX or DLL on NT.
    1. If authentication fails, return an error to ReportCast Server.
    2. If authentication succeeds, pass the request to the e.Reporting Server.
8. The e.Reporting Server processes the user login and password.
    1. If login fails, the e.Reporting Server returns an error to ReportCast.
    2. If login succeeds, the e.Reporting Server returns successful to ReportCast.
9. If login fails:
    1. ReportCast returns the AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE, if set, or the HTTP 401 status code to the web server.
    2. The web server returns the error status to the browser.
    3. The browser displays Login, and the process begins again.

The following diagram shows the ReportCast Agent and ReportCast Server Security Extension authentication process in more detail.


5. or 7. Incoming request
ReportCast Agent or ReportCast Server
RCASE/ RCSE Enabled?
No
Yes
Call custom user authentication function
User authenticated?
No
Return status
Yes
Return success

Developing the ReportCast Security Extension

To override the default authentication process and enable the ReportCast Agent Security Extension or ReportCast Server Security Extension, you develop a custom DLL to contain the RCASE or RCSE security library and set a registry key on NT or an environment variable on UNIX.

ReportCast Agent Security Extension library functions

When the ReportCast Agent Security Extension is enabled, ReportCast calls the RCASE library functions shown in the following table.

RCASE function	Description
AcRCAgentAuthenticateRequest Body	Authenticate the request data.
AcRCAgentAuthenticateRequest Header	Authenticate or filter the HTTP request headers and put the filter data into a buffer.

If AcRCAgentAuthenticateRequestHeader does not modify the HTTP request headers or if RCASE is disabled, ReportCast Agent sends all the HTTP request headers to ReportCast Server.

If AcRCAgentAuthenticateRequestBody does not modify the HTTP request data or if RCASE is disabled, ReportCast Agent sends the original request body to ReportCast Server.

The following section describes the AcRCAgentAuthenticatedRequestBody and AcRCAgentAuthenticatedRequestHeader functions.

AcRCAgentAuthenticateRequestBody

Use AcRCAgentAuthenticateRequestBody() to authenticate the HTTP request data. When RCASE is enabled, ReportCast calls this function after calling AcRCAgentAuthenticateRequestHeader.

Syntax

```
int AcRCAgentAuthenticateRequestBody ( const char *httpRequest, char
    *filteredHttpRequest, int *size )
```

Parameter

const char *httpRequest
Input. The request data sent by the web browser.

char *filteredHttpRequest
Output. The filtered request modified by AcRCAgentAuthenticateRequestBody().

int *size

Input/Output. The size of the buffer filteredHttpRequest. Use AcRCAgentAuthenticateRequestBody() to modify the value of size to indicate the actual size of the filtered HTTP body.

The following are the values for size.

Value of size	Description
0	The request body was not modified.
<= size	The new request body is in FilteredHttpRequest. The value of the parameter size is the length of FilteredHttpRequest.
> size	The function requires a larger buffer.

Description

Authenticate or filter the request data, and place the filtered data into the buffer filteredHttpRequest. If AcRCAgentAuthenticateRequestBody() does not modify the request body, ReportCast sends the original request body to ReportCast Server.

Do not write outside the buffer filteredHttpRequest. Doing so corrupts memory, which eventually leads to a crash. The parameter size indicates the amount of space allocated for filteredHttpRequest.

Returns

ReportCast expects AcRCAgentAuthenticateRequestBody() to return the following values.

Return value	Description
0	Success. ReportCast Agent can send the data to ReportCast Server.
1	Failure. ReportCast Agent sends an error page to the web browser. ReportCast uses either the 401 status code, or the value of the NT registry key or UNIX environment variable AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE.
2	Increase the buffer size. The parameter size contains the new value for the size of the filteredRequest buffer. ReportCast Agent allocates filteredRequest with the value of size and repeats the request. The new size must be larger than the original size.

Use any other return value to indicate that an unexpected error occurred in your implementation of AcRCAgentAuthenticateRequestBody().

AcRCAgentAuthenticateRequestHeader

Use AcRCAgentAuthenticateRequestHeader() to authenticate or filter the HTTP request headers and place the filtered headers into the buffer filteredHttpHeaders. If RCASE is enabled, ReportCast calls this function before calling AcRCAgentAuthenticateRequestBody.

Syntax

```
int AcRCAgentAuthenticateRequestHeader ( const char **httpHeaders, int
    *numberOfHeaders, char *filteredHttpHeaders, int *size )
```

Parameter

const char **httpHeaders

Input. An array of null-terminated strings, each representing an HTTP header.

int *numberOfHeaders

Input/Output. Used for input, numberOfHeaders contains the number of elements in the httpHeaders array. Used for output, numberOfHeaders contains the number of headers stored in the buffer FilteredHttpHeaders.

char *filteredHttpHeaders

Output. A buffer containing the filtered headers separated by a null character.

int *size

Input/Output. The size of the buffer filteredHttpHeaders. Used for input, size contains the size of filteredHttpHeaders. Used for output, size contains one of the following values.

Value of size	Description
0	The request headers were not modified.
<= size	The new request headers are in FilteredHttpHeaders. The value of the parameter size is the length of filteredHttpHeaders.
> size	The function requires a larger buffer.

Use any other return value to indicate that an unexpected error occurred in your implementation of AcRCAgentAuthenticateRequestHeader.

Description

Use AcRCAgentAuthenticateRequestHeader() to authenticate or filter the request headers and place the filtered headers into the buffer filteredHttpHeaders. If AcRCAgentAuthenticateRequestHeader() does not modify the request headers, ReportCast sends the original request headers to ReportCast Server.

Do not write outside the buffer filteredHttpHeaders. Doing so corrupts memory, which eventually leads to a crash. The parameter size indicates the amount of space allocated to filteredHttpHeaders.

Returns ReportCast expects AcRCAgentAuthenticateRequestHeader() to return the following values.

Return value	Description
0	Success. ReportCast Agent can send the data to ReportCast Server.
1	Failure. ReportCast Agent sends an error page to the web browser. ReportCast uses either the 401 status code or the value of the NT registry key or UNIX environment variable AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE.
2	Increase the buffer size. The parameter size contains the new value for the size of the filteredHttpHeaders buffer. ReportCast Agent allocates filteredHttpHeaders with the value of size and repeats the request. The new size must be larger than the original size.

Example

```
#ifdef WIN32
#include <windows.h>
#define EXPORT WINAPI
#else
#define EXPORT
#endif

#ifdef __cplusplus
extern "C" {
#endif

int EXPORT
AcRCAgentAuthenticateRequestHeader(const char ** httpHeaders, int*
numberOfHeaders, char* filteredHttpHeaders, int* size) {

/* Your code here */

}

int EXPORT
AcRCAgentAuthenticateRequestBody(const char *httpRequest, char*
filteredHttpRequest, int* size) {

/* Your code here */

}

#ifdef __cplusplus
}
#endif
```

ReportCast Server Security Extension library functions

When the ReportCast Server Security Extension is enabled, ReportCast calls the RCSE library functions shown in the following table.

RCSE function	Description
AcWebSecurityStart	Initialize the RCSE library.
AcWebSecurityAuthenticate	Authenticate a user by translating user security information into e.Reporting Server login information.
AcWebSecurityAuthenticateEx	Specify the extended credentials to use to log a user into the e.Reporting Server.
AcWebSecurityStop	Shut down the RCSE library.

If the AcWebSecurityStart() or AcWebSecurityStop() functions are not present in the RCSE library, ReportCast continues and does not issue an error message. AcWebSecurityAuthenticate() must be present in the RCSE library if the ReportCast Security Extension is enabled. If AcWebSecurityAuthenticateEx() is present in the RCSE library, ReportCast calls it instead of AcWebSecurityAuthenticate().

For more information about enabling the ReportCast Security Extension, see "Enabling RCASE and RCSE," later in this chapter.

For information about setting the security variables, see "Setting the ReportCast security variables," earlier in this chapter.

The following section describes the RCSE library functions. The section lists the functions in the order in which you use them.

AcWebSecurityStart function

This function initializes the ReportCast security library.

Syntax void AcWebSecurityStart( )

Description AcWebSecurityStart() performs internal initialization, such as loading tables. If AcWebSecurityStart() is present in the RCSE library, ReportCast calls it once, when it loads the security library.

AcWebSecurityAuthenticate function

This function authenticates a user.

Syntax

```
int AcWebSecurityAuthenticate(char **httpHeaders, int numberOfHeaders,
   char *username, char *password)
```

Description

AcWebSecurityAuthenticate() is called each time ReportCast connects to an e.Reporting Server to service a request. If this function is not present in the RCSE library or an error occurs during the operating system call to locate the function, ReportCast sends an HTML error page generated from the Error.achtml template to the browser.

This function supports using the HTTP headers from the browser to determine the user name and password that ReportCast uses to log on to the e.Reporting Server.

Parameter

httpHeaders [in]
An array of null-terminated strings, each representing an HTTP header using the format defined in the HTTP 1.1 specification.

numberOfHeaders [in]
Number of elements in the httpHeaders array.

username [out]
Name ReportCast uses to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the user name into this location.

password [out]
Password to use to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the password into this location.

Returns

0 on success, 1 on failure.

If AcWebSecurityAuthenticate() returns 1, ReportCast does not attempt to log on to the e.Reporting Server and returns to the web browser a status code of 401 or the value specified by the AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE variable.

AcWebAuthenticateEx function

This function specifies the extended credentials to use to log a user into the e.Reporting Server.

Syntax

```
int AcWebSecurityAuthenticateEx (
    char **headers,
    int numHeaders,
    char *username,
    char *password,
    char *extendedCredentialsBuffer,
    int *extendedCredentialsSize )
```

Description

If AcWebSecurityAuthenticateEx() is present in the ReportCast Security Extension, ReportCast calls AcWebSecurityAuthenticateEx() instead of AcWebSecurityAuthenticate(). Use AcWebSecurityAuthenticateEx() to specify any extended security credentials required to log a user into the e.Reporting Server. Use this function if you are building an LDAP security scheme for your e.Reporting Server and ReportCast.

Parameter

char **headers [in]
An array of null-terminated strings, each representing an HTTP header using the format defined in the HTTP 1.1 specification.

int numHeaders [in]
The size of the array contained in the headers parameter.

char *username [out]
Name ReportCast uses to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the user name into this location.

char *password [out]
Password to use to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the password into this location.

char *extendedCredentialsBuffer [out]
The buffer to receive the set of additional credentials to use to log into the e.Reporting Server. ReportCast passes in the buffer to which extendedCredentialsBuffer points.

int *extendedCredentialsSize [in/out]
The length of the extendedCredentialsBuffer before ReportCast calls the AcWebSecurityAuthenticateEx() function. After AcWebSecurityAuthenticateEx() returns, extendedCredentialsSize points to the number of bytes in the buffer that have been filled with extended credentials.

AcWebSecurityAuthenticateEx() must specify a non-empty value for at least the user name.

Returns 0 on success.

1 on failure.

2 if the extendedCredentialsBuffer is not long enough to store the extended credentials for the user.

AcWebSecurityStop function

This function shuts down the ReportCast security library.

Syntax void AcWebSecurityStop( )

Description Performs internal cleanup for the RCSE library. If AcWebSecurityStop() is present in the library, it is called once, when ReportCast shuts down.

Development considerations

Following are some items to consider when developing the Actuate ReportCast Security Extension:

- The functions that the security library exports, AcWebSecurityStart(), AcWebSecurityAuthenticate(), and AcWebSecurityStop(), should all be declared extern C if you are writing the library in C++. This procedure prevents the compiler from changing the symbol name into something other than what ReportCast is looking for.
- Your security library must be thread-safe and cannot depend on any one thread handling a particular request.
- The HTTP header names that are passed to the security library should be in the format specified in the HTTP specification. The headers do not use the CGI or ISAPI format. For example, the header Accept-Charset is used, rather than HTTP_ACCEPT_CHARSET. This practice supports writing the Security Extension code using the standard HTTP headers so that it is portable among different web servers and platforms.
- Ensure that your code is not case-sensitive.
- When testing your ReportCast Security Extension library, be aware that ReportCast caches connections to the e.Reporting Server. If you are changing user passwords on the server during your testing, you might have to clear cached connections using the ReportCast administration page.

Example The following code example shows a template used to develop a security library:

```
#ifdef WIN32
#include <windows.h>
#endif

#ifdef __cplusplus
extern "C" {
#endif

void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStart() {
// Your code here.
}
void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStop() {
// Your code here.
}

int
#ifdef WIN32
WINAPI
#endif
AcWebSecurityAuthenticate(char **httpHeaders,
   int numberOfHeaders,
   char *username,
   char *password) {
// Your code here.
}

#ifdef __cplusplus
}
#endif
```

NT-specific information

On NT, you must use the WINAPI calling convention in the function definitions for the RCSE functions. Declare the RCSE functions in a DEF file that is linked to the DLL.

For example, the following code shows the contents of a DEF file to link into the DLL:

```
DESCRIPTION 'Your Security Extension Description'
EXPORTS
   AcRCAgentAuthenticateRequestBody
   AcRCAgentAuthenticateRequestHeader
   AcWebSecurityStart
   AcWebSecurityStop
   AcWebSecurityAuthenticate
```

Server-specific information

The following sections contain additional information that applies to the supported web servers, Netscape and Microsoft.

Netscape server

The ReportCast Agent Security Extension supersedes any previous ReportCast authentication customization solutions. The security extension feature allows implementation of custom security authentication that is portable and avoids the need to add directives to the Obj.conf file to achieve customization.

Microsoft Internet Information server

By default, the Microsoft Internet Information server (IIS) uses the Windows NT security information from the browser to grant or deny access to all resources on the web server. IIS considers the ReportCast DLL to be the resource to which it must authorize access, rather than the e.Reporting Server. Each user name and password must exist on both the NT system to access ReportCast, and the e.Reporting Server to access the Report Encyclopedia. Because this setup is not practical in some cases, the ReportCast configuration customizes this behavior.

If the security extension is not enabled, ReportCast overrides the default behavior of IIS. A component of ReportCast acts as an authentication filter. This filter instructs IIS to access ReportCast as the NT user designated for IIS anonymous access. In this configuration, IIS must be configured to allow anonymous access. Windows NT Challenge/Response must be disabled. When ReportCast attempts to connect to the e.Reporting Server, it uses Basic Authentication to gather the user name and password information.

The alternative authentication method is to use the ReportCast Agent Security Extension. If the security extension is enabled, ReportCast does not override the default behavior of IIS to authorize access to the ReportCast DLL. Authentication information for the ReportCast DLL comes from the browser. ReportCast does not filter the user information. The user information coming from the browser must identify a valid NT user. If the browser does not send authentication information, and if IIS is configured to allow anonymous access, IIS accesses ReportCast as the NT user designated for anonymous access.

If you want to access ReportCast using a different Windows NT user than that specified by the browser, install an additional ISAPI filter DLL in Internet Information Server.

Depending on the security model you use, the security extension returns the same user information as the browser sent or maps the user information to a completely different user. In either case, the user the security extension returns must exist in the e.Reporting Server if ReportCast is to log on successfully.

Enabling RCASE and RCSE

Set AC_REPORTCAST_SECURITY_LIBRARY for each RCASE and RCSE for which you created a security library. Setting AC_REPORTCAST_SECURITY_LIBRARY allows ReportCast Agent and ReportCast Server to locate your security libraries. If this variable is not set, ReportCast Agent and ReportCast Server use the default authentication scheme.

The variable value is the name of the DLL or shared object file that contains the authentication functions. The name can include an absolute or relative path.

ReportCast loads the library once, the first time a request is made to ReportCast. ReportCast unloads the library when it exits. If ReportCast cannot load the library, ReportCast sends an error page to the browser.

You must enable each RCASE and RCSE. If you have multiple ReportCast Agent Security Extensions, enabling RCASE on one machine enables only that RCASE and not any others. If you have multiple ReportCast Server Security Extensions, enabling RCSE on one machine enables only that RCSE and not any others.

If you installed ReportCast Agent and ReportCast Server on the same UNIX machine, set AC_REPORTCAST_SECURITY_LIBRARY both in the web server startup script and in the ReportCast Server startup script.

Use regedit to enable RCASE and RCSE on NT. For more information about running Regedit.exe and creating registry keys, see "Setting the ReportCast security variables," earlier in this chapter.

How to set the RCASE security variable on NT

On each NT machine on which an RCASE resides do the following:

1 Choose Start➛Run.

2 In Run, type:

```
regedit.exe
```

3 Choose OK.

4. In Registry Editor, navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.
5. Create a new registry key, AC_REPORTCAST_SECURITY_LIBRARY.
6. Set the AC_REPORTCAST_SECURITY_LIBRARY variable to the full path name of the ReportCast Agent Security Extension library's location.

How to set the RCASE security variable on UNIX

On each UNIX machine on which an RCASE resides, type:

```
setenv AC_REPORTCAST_SECURITY_LIBRARY <RCASE library location>
```

where RCASE library location is the full path name of the ReportCast Agent Security Extension library's location.

If ReportCast Agent and ReportCast Server reside on the same machine, set AC_REPORTCAST_SECURITY_LIBRARY in the web server's startup script.

RCASE is now enabled.

How to set the RCSE security variable on NT

On each NT machine on which an RCSE resides, do the following:

1. Choose Start➛Run.
2. In Run, type:

   ```
   regedit.exe
   ```

3. Choose OK.
4. In Registry Editor, navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.
5. Create a new registry key, AC_REPORTCAST_SECURITY_LIBRARY.
6. Set the AC_REPORTCAST_SECURITY_LIBRARY variable to the full path name of the ReportCast Security Extension library's location.

How to set the RCSE security variable on UNIX

On each UNIX machine on which an RCSE resides, type:

```
setenv AC_REPORTCAST_SECURITY_LIBRARY <RCSE library location>
```

where RCSE library location is the full path name of the ReportCast Security Extension library's location.

If ReportCast Agent and ReportCast Server reside on the same machine, set AC_REPORTCAST_SECURITY_LIBRARY in the ReportCast Server startup script, startreportcast.sh.

RCSE is now enabled.

Chapter

7

Security for Internet deployment

This chapter contains the following topics:

- About network security and Actuate
- Data protection strategies
- Understanding Actuate component communication
- Deploying Actuate products in Internet environments

About network security and Actuate

This chapter discusses configuring Actuate e.Reporting Suite 5 products for deployment in a secured network environment. Project managers, installation engineers, and security team members can plan hardware and software requirements, installation configuration options, and how Actuate fits into the Internet security infrastructure of their organization using information from this chapter. The topics discussed include the technologies that deter unauthorized access to corporate data, such as firewalls, Network Address Translation (NAT), and proxy servers.

This chapter is not a guide to securing your network. It is a general guide to how Actuate products work with secure environments. This chapter assumes a working knowledge of TCP/IP communications.

Data protection strategies

Actuate e.Reporting Suite 5 provides a structured content generation solution for web applications. Deploying Actuate applications developed for the Internet requires planning for network security.

Internet applications support access to information within an organization from outside that organization. Because the organization's internal network is connected to the Internet, there is the risk of unauthorized access to the corporate network and to the data that resides on that network.

Organizations use one or a combination of the technologies described in the following sections to prevent unauthorized access to the corporate network.

Protecting corporate data using firewalls

Typically companies use firewalls to prevent unauthorized access to corporate networks and data.

A firewall is a system or group of systems that restrict access between two networks, such as an organization's internal network and the Internet. Firewalls keep unauthorized users out. As a result, firewalls prevent damage caused by malicious programs such as worms and viruses from spreading to other parts of your network. At the same time, firewalls allow legitimate business to tunnel through the firewall and be efficiently conducted on your network.

Firewalls can be used to restrict access between two internal networks, for example, the accounting and engineering networks. Security teams configure firewalls to allow traffic using specific protocols, such as HTTP, over specific network addresses and ports.

Protecting corporate data using Network Address Translation

Companies also use Network Address Translation (NAT). NAT routers and software support private networks using unregistered, private IP (Internet Protocol) addresses to connect to the Internet.

Protecting corporate data using proxy servers

Proxy servers, specialized web servers or hardware that operate on or behind a firewall, improve efficient use of network bandwidth and offer enhanced network security.

Understanding Actuate component communication

This section describes the relationship between Actuate component communication and your secure network environment.

Actuate e.Reporting Server, ReportCast, and all client products use TCP/IP and related protocols for all network communication. The communications process between client products and the e.Reporting Server is the same as the communications process between ReportCast and the e.Reporting Server.

The originating, or outbound, port is the port that a component uses to initiate two-way connections between itself and another component. The destination, or inbound, port is the port to which an originating port initiates two-way connections.

Actuate components perform the following communications tasks:

- Communicating between the web browser and web server
- Establishing the e.Reporting Server port
- Connecting to and browsing the Report Encyclopedia
- Running e.reports
- Viewing e.reports in DHTML format
- Viewing e.reports using the LRX

You configure the ports to use for specific communications tasks at e.Reporting Server and ReportCast startup using registry settings on NT systems or environment variables on UNIX systems.

The types of ports you can configure are:

- e.Reporting Server port. This port is used for the ReportCast to e.Reporting Server connection.
- Cached connection management. This port is used to control the number of ReportCast's cached connections.
- View process port. This port is used for viewing DHTML reports, PDF content, and XML data and XML display.
- Data reply connection. This port is used to view reports using the LRX or Viewer, retrieving parameters for running reports, and notifications for report requests.

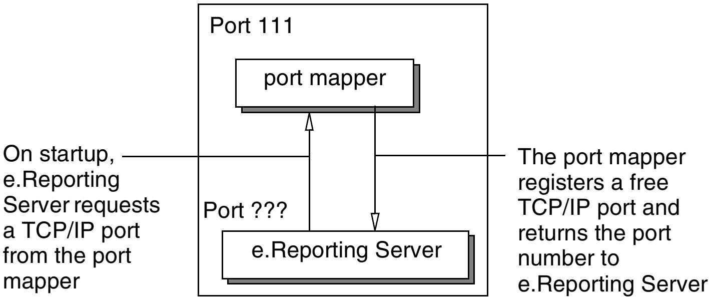
Establishing the e.Reporting Server port

The e.Reporting Server port is the destination port for connections from ReportCast to the e.Reporting Server. ReportCast and the e.Reporting Server use the connection for:

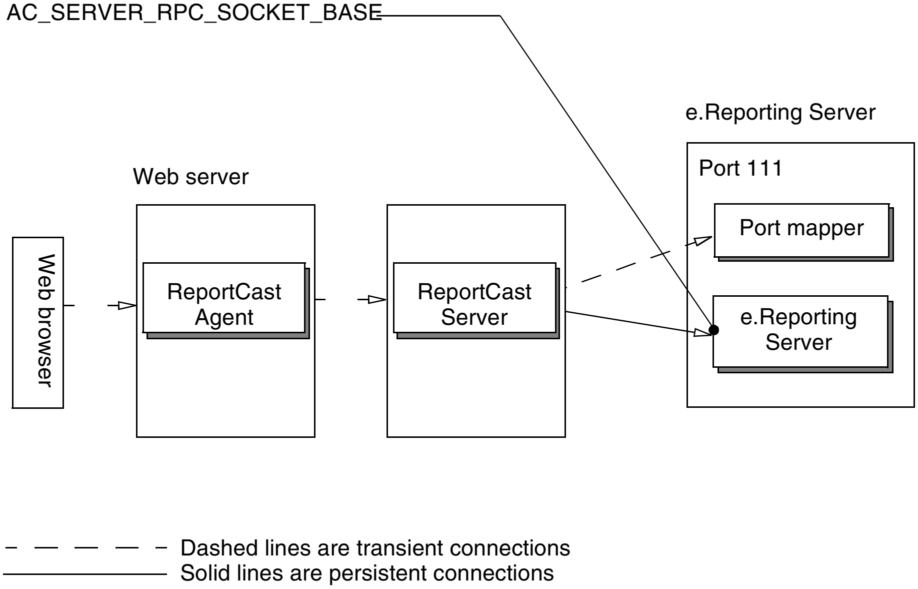
- Login
- Report Encyclopedia browsing
- Report Encyclopedia administration
- Initiating report requests
- Initiating report viewing

e.Reporting Server applications use the TCP/IP port mapper to determine the port through which to communicate with the e.Reporting Server. When the e.Reporting Server starts, it requests a communications port from the port mapper. The port mapper locates an available TCP/IP port, registers the port number, and returns the port number to the e.Reporting Server.

The port mapper listens for queries on port 111. Applications query the port mapper for the port used by the e.Reporting Server. The port mapper responds with the registered port. Applications use the registered port to connect to the e.Reporting Server.


Port 111
port mapper
On startup, e.Reporting Server requests a TCP/IP port from the port mapper
Port ???
e.Reporting Server
The port mapper registers a free TCP/IP port and returns the port number to e.Reporting Server

The following illustration shows the connection between ReportCast and the e.Reporting Server.



The e.Reporting Server port uses the RPC protocol over TCP. There is one destination port for all user connections. You can configure the number of cached connections. The peak number of connections depends on the user load.

Configuring the e.Reporting Server port

You can set the port number to use for connections and the number of ports to search if that port is not available.

To configure the e.Reporting Server port, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- AC_SERVER_RPC_SOCKET_BASE sets the port number that the e.Reporting Server attempts to register with the port mapper.
- AC_SERVER_RPC_SOCKET_COUNT sets the number of ports search if the registered port is not available.

When the e.Reporting Server attempts to register a port with the port mapper, it searches from AC_SERVER_RPC_SOCKET_BASE to AC_SERVER_RPC_SOCKET_BASE + AC_SERVER_RPC_SOCKET_COUNT-1 to find an open port. If no ports are available within the configured range, the port mapper allocates a port. The default is to allow the port mapper to allocate the port.

For detailed information about configuring registry keys on NT or environment variables on UNIX, see Chapter 2, “Report server administration on UNIX,” or Chapter 3, “Report server administration on Windows servers,” in the *Actuate e.Reporting Server Guide*.

Configuring cached connection management

ReportCast caches connections to the e.Reporting Server port, AC_SERVER_RPC_SOCKET_BASE. To control the number of cached connections, set the following registry keys on NT or environment variables on UNIX on the web server machine:

- AC_REPORTCAST_CONNECTION_CACHE_SIZE limits the total number of connections. Actuate drops the oldest connections automatically.
- AC_REPORTCAST_CONNECTION_CACHE_TIMEOUT limits the life of the connection.

For more information about configuring cached connections, see Chapter 2, “Administering ReportCast.”

Web browser to web server communication

Web browser to web server communication uses the standard HTTP or HTTPS (SSL) protocols. Actuate products do not require any special configuration or consideration for HTTP or HTTPS communication. HTTPS is transparent to Actuate.

Connecting to and browsing the Report Encyclopedia

The following refers to ReportCast, rather than ReportCast Agent and ReportCast Server, in order to simplify discussion. The ReportCast process is:

1. A user makes a request using a web browser.
2. The web browser passes the request to the web server.
3. The web server recognizes the acweb keyword in the request and passes it to ReportCast Agent.
4. ReportCast Agent passes the request to ReportCast Server.
5. ReportCast Server processes the request for the e.Reporting Server and passes it to the e.Reporting Server.
6. The e.Reporting Server processes the request and sends the results to ReportCast Server.

7. ReportCast Server prepares a web page with the results and sends it to ReportCast Agent.
8. ReportCast Agent sends the web page to the web server.
9. The web server passes the web page to the web browser.

ReportCast Agent and ReportCast Server may also call their security extensions to authenticate user access to the e.Reporting Server, depending upon the site's configuration and requirements. See Chapter 6, "Internal ReportCast security," for more information about the ReportCast security extensions.

The process for connecting to and browsing the Report Encyclopedia is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder.
- On the first connection, ReportCast opens an originating port.
- ReportCast connects to the port mapper using TCP to determine the e.Reporting Server port.
- The port mapper returns the e.Reporting Server port.
- ReportCast connects to the e.Reporting Server using the destination port supplied by the port mapper. ReportCast uses the new connection to retrieve information from the e.Reporting Server. All users share one destination port.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

Connections from the web browser to the web server and ReportCast, and from ReportCast to the port mapper, are transient. The connection between ReportCast and the e.Reporting Server remains open.

The desktop products use the same process to communicate with the e.Reporting Server.

Running and viewing reports

This section describes the process of running and viewing reports and how to configure the view process and data reply ports.

Running reports

One connection is used for each concurrent user running a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for running reports is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80.
- ReportCast requests the parameters needed to run the report.
- The e.Reporting Server establishes a new connection to ReportCast for sending the report's parameter data.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The desktop products use the same process to run a report.

The process for running reports uses data reply ports. Configuring data reply ports is described in "Configuring data reply ports," later in this chapter.

Configuring data reply ports

The LRX and Viewer use the data reply ports to view reports, to retrieve parameters for running reports, and for report request notification.

The data reply connection process is:

- ReportCast logs in and gets an RPC connection to the e.Reporting Server.
- User initiates a request to run a report or to view content using the LRX viewer.
- ReportCast requests parameters from the e.Reporting Server or requests content for the LRX viewer.
- e.Reporting Server acknowledges the request and initiates a connection to ReportCast:
  - e.Reporting Server opens an originating port (AC_SERVER_REPLY_SOCKET_BASE).
  - e.Reporting Server connects to the destination port on the ReportCast machine (AC_SERVER_SOCKET_BASE).
- e.Reporting Server sends the report data.
- e.Reporting Server closes the connection.

To configure originating data reply ports, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- AC_SERVER_REPLY_SOCKET_BASE sets the number at which to begin allocating originating ports.
- AC_SERVER_REPLY_SOCKET_COUNT sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from AC_SERVER_REPLY_SOCKET_BASE to AC_SERVER_REPLY_SOCKET_BASE + AC_SERVER_REPLY_SOCKET_COUNT-1 to find an open port. The default value for AC_SERVER_REPLY_SOCKET_BASE is 4096. The default value for AC_SERVER_REPLY_SOCKET_COUNT is 8192.

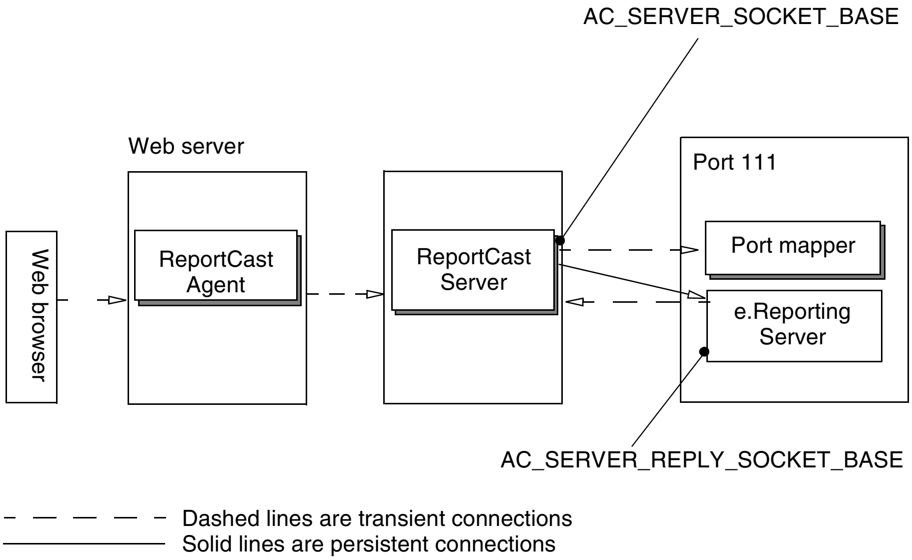
To configure destination ports, set the following registry keys (on NT) or environment variables (on UNIX) on the ReportCast or Viewer machine:

- AC_SERVER_SOCKET_BASE sets the number at which to begin allocating destination ports.
- AC_SERVER_SOCKET_COUNT sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from AC_SERVER_SOCKET_BASE to AC_SERVER_SOCKET_BASE + AC_SERVER_SOCKET_COUNT-1 to find an open port. The default value for AC_SERVER_SOCKET_BASE is 4096. The default value for AC_SERVER_SOCKET_COUNT is 8192.

Data reply ports use the TCP protocol and the UDP protocol for notifications. Actuate opens one port for each concurrent data request. For example, if ten users simultaneously request a report, ten pairs of ports open. For each pair of ports, one is on the origination side and one is on the destination side. Data reply connections are opened and closed for each data reply request.

The following illustration shows the connections between ReportCast and the e.Reporting Server for data reply requests.



Viewing e.reports as DHTML, PDF, or XML

Viewing DHTML, PDF, or other structured content using the view process uses one port per view process. Each view process port is shared by all users.

The process for viewing e.reports in DHTML format is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view a report as DHTML.
- ReportCast makes a request to view a DHTML report.
- The e.Reporting Server tells ReportCast which port to use to retrieve the DHTML content.
- ReportCast connects to the e.Reporting Server port for viewing DHTML content. The e.Reporting Server returns DHTML content to ReportCast. Actuate uses one port for each view process. Typically, the number of ports used is one or two. See the *Actuate e.Reporting Server Guide* for more information about e.Reporting Server ports. All users share the same view process ports.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The process of viewing e.reports in DHTML, PDF, or XML uses the view process port. Configuring the view process port is described in "Configuring the view process port," later in this chapter.

Configuring the view process port

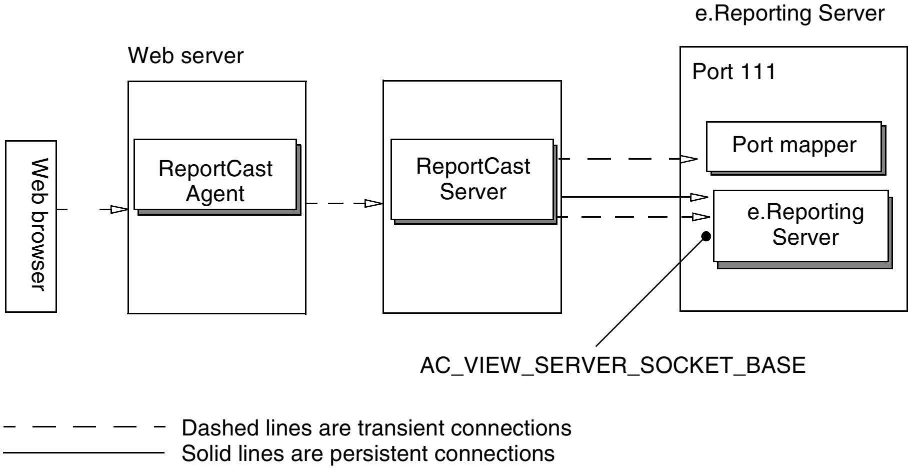
The view process port is used for viewing DHTML, PDF, and XML content. Set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- AC_VIEW_SERVER_SOCKET_BASE sets the port number at which to begin allocating view process connection ports. The default value is 15500.
- AC_VIEW_SERVER_SOCKET_COUNT sets the maximum number of ports to search for open ports. The default value is 200.

When the e.Reporting Server receives a request for the view process, it searches from AC_VIEW_SERVER_SOCKET_BASE to AC_VIEW_SERVER_SOCKET_BASE + AC_VIEW_SERVER_SOCKET_COUNT-1 to find an open port. Set AC_VIEW_SERVER_SOCKET_BASE to a number greater than 1024. Set AC_VIEW_SERVER_SOCKET_COUNT to a number less than 65535.

The view process port uses the TCP protocol. Actuate uses one port for each view process. View process connections are opened and closed for each request to view DHTML, XML, or PDF content.

The following illustration shows the connections among ReportCast, the view process, and the e.Reporting Server.



Viewing e.reports using the LRX

One connection is used for each concurrent user viewing a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for viewing e.reports using the LRX is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with the request to view the reporting using the LRX.
- ReportCast makes a request to view a report using the LRX.
- The e.Reporting Server establishes a new connection to a ReportCast port for sending report data for viewing.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The Viewer and other desktop products use the same process to view a report on the e.Reporting Server.

The process for viewing e.reports using the LRX uses data reply ports. Configuring data reply ports is described in “Configuring data reply ports,” earlier in this chapter.

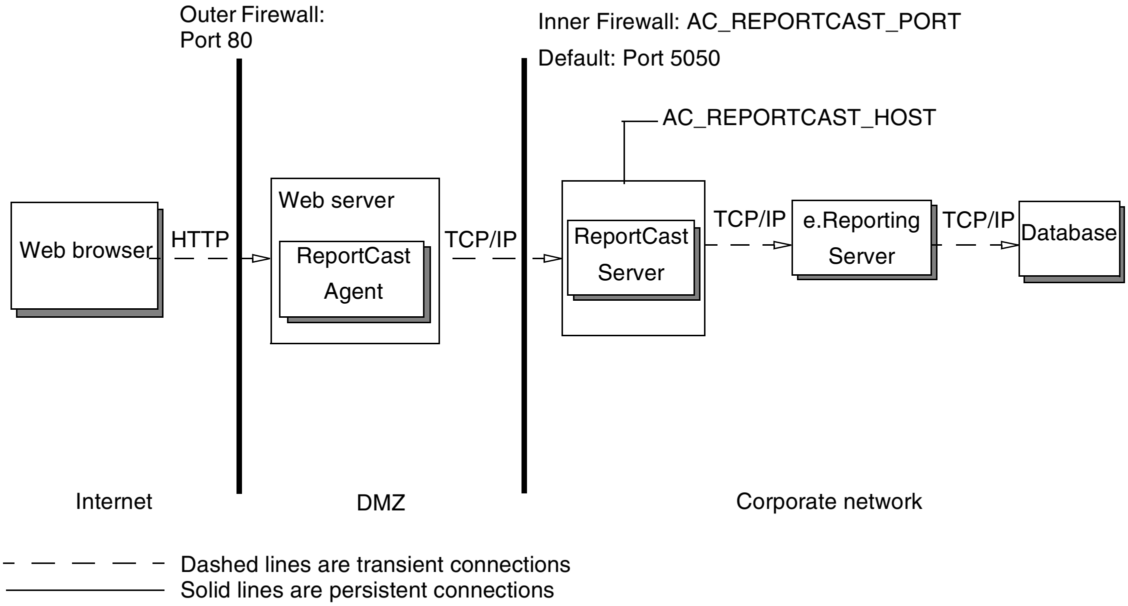
Connecting to the Report Encyclopedia with the split ReportCast configuration

The split ReportCast configuration supports installing the ReportCast Agent on one machine, and the ReportCast Server on another machine. You can simplify the configuration of your firewall by placing the ReportCast Agent with the web server in the DMZ and the ReportCast Server on a separate machine on the corporate net.

The split ReportCast configuration provides the following advantages:

- Requires only one port through the inner firewall. The default is port 5050. The port is configurable.
- Does not require a proxy server.

The following illustration shows the split ReportCast configuration.



The split ReportCast communication process is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder, run a report, or view a report.
- The web server passes the request to the ReportCast Agent.
- The ReportCast Agent connects to the ReportCast Server on the port specified by AC_REPORTCAST_PORT on the machine specified by AC_REPORTCAST_HOST and passes the request to the ReportCast Server.
- The ReportCast Server connects to the e.Reporting Server.
- The ReportCast Server generates the web reporting page and returns it to the ReportCast Agent.
- The ReportCast Agent returns the web reporting page to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

Configuring the split ReportCast configuration

To configure the split ReportCast configuration, set the following environment variables:

- AC_REPORTCAST_PORT sets the port to use for communicating between the ReportCast Agent and the ReportCast Server. Set AC_REPORTCAST_PORT on both the ReportCast Agent and ReportCast Server machines. The port number must match on both machines.
- AC_REPORTCAST_HOST sets the name of the machine hosting the ReportCast Server. Set AC_REPORTCAST_HOST only on the ReportCast Agent machine.

When the ReportCast Agent receives a request, it connects to the ReportCast Server on the port specified by the AC_REPORTCAST_PORT setting, on the host machine specified by the AC_REPORTCAST_HOST setting.

The default value for AC_REPORTCAST_PORT is 5050. Set AC_REPORTCAST_PORT to the same value on both the ReportCast Agent and the ReportCast Server machines.

Set AC_REPORTCAST_PORT as follows:

- For Netscape web servers only, on the ReportCast Agent side, set AC_REPORTCAST_PORT in the Netscape start script.
- For all web servers including Netscape on the ReportCast Agent side, set AC_REPORTCAST_PORT in the nph-actuate.cgi and ph-actuate.cgi scripts.
- On the ReportCast Server side, set AC_REPORTCAST_PORT in startreportcast.sh.

Set AC_REPORTCAST_HOST on the ReportCast Agent side only as follows:

- For Netscape web servers only, set AC_REPORTCAST_HOST in the Netscape start script.
- For all web servers, including Netscape, set AC_REPORTCAST_HOST in the nph-actuate.cgi and ph-actuate.cgi scripts.

Deploying Actuate products in Internet environments

There are many options for configuring network security. This section presents some scenarios and discusses requirements for deploying Actuate products in secure Internet environments.

Some routers and firewalls perform Network Address Translation (NAT). You can set up NAT to perform one-to-one IP address mapping in both directions. In a connection from point A to point B, NAT must map both A to B, and B to A.

The following are sample network security configuration options:

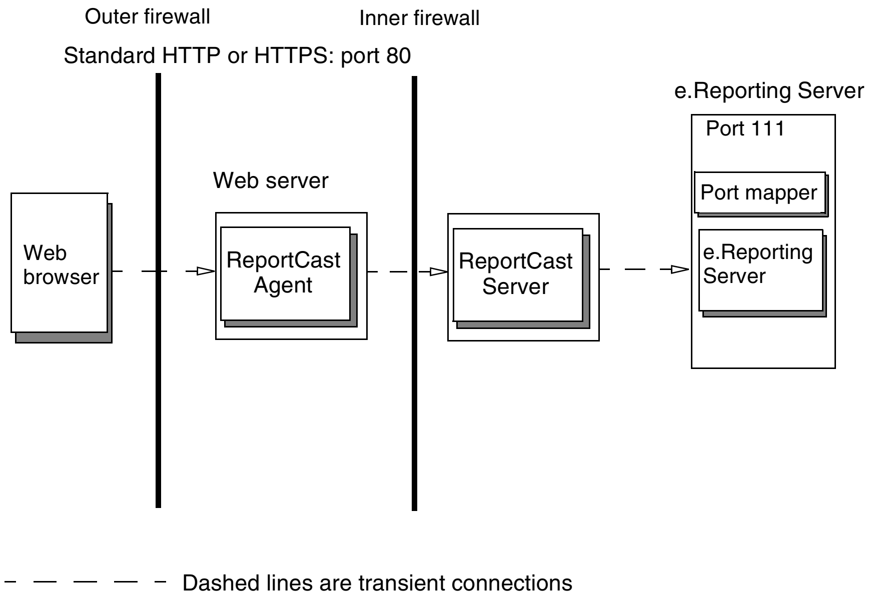
- Split ReportCast installation places the web server and ReportCast Agent in the DMZ. The ReportCast Server and e.Reporting Server reside behind the second firewall. This configuration is recommended.
- Web server and ReportCast in the DMZ places the web server and ReportCast between two firewalls. The e.Reporting Server resides behind the inner firewall.
- Proxy server in the DMZ places a proxy server between two firewalls. The web server, ReportCast, and e.Reporting Server reside behind the inner firewall.

The following sections discuss firewall configuration options, including port configuration instructions.

Understanding configuration options and firewall setup

Most firewall configurations are set up to accept traffic from specific protocols, such as HTTP or HTTPS traffic only. Generally, corporations expose only limited ports and one IP address to the World Wide Web. All Internet traffic comes through specific ports, usually including port 80, and to the one outside IP address. From port 80, Internet traffic is routed to appropriate internal IP addresses and ports, often using Network Address Translation (NAT).

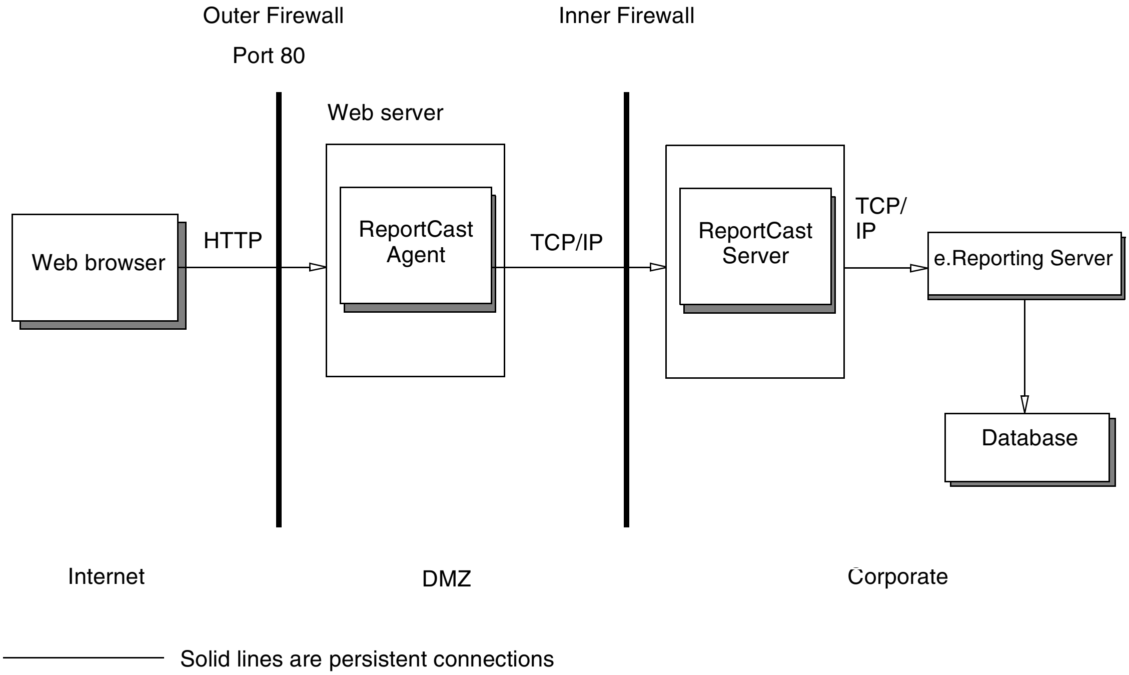
The web browser to web server communications process is the same for each configuration. If there is a firewall, communication uses standard HTTP or HTTPS protocols, as shown in the following diagram.



The following sections discuss each firewall configuration option.

Dual firewalls with ReportCast Agent in the DMZ

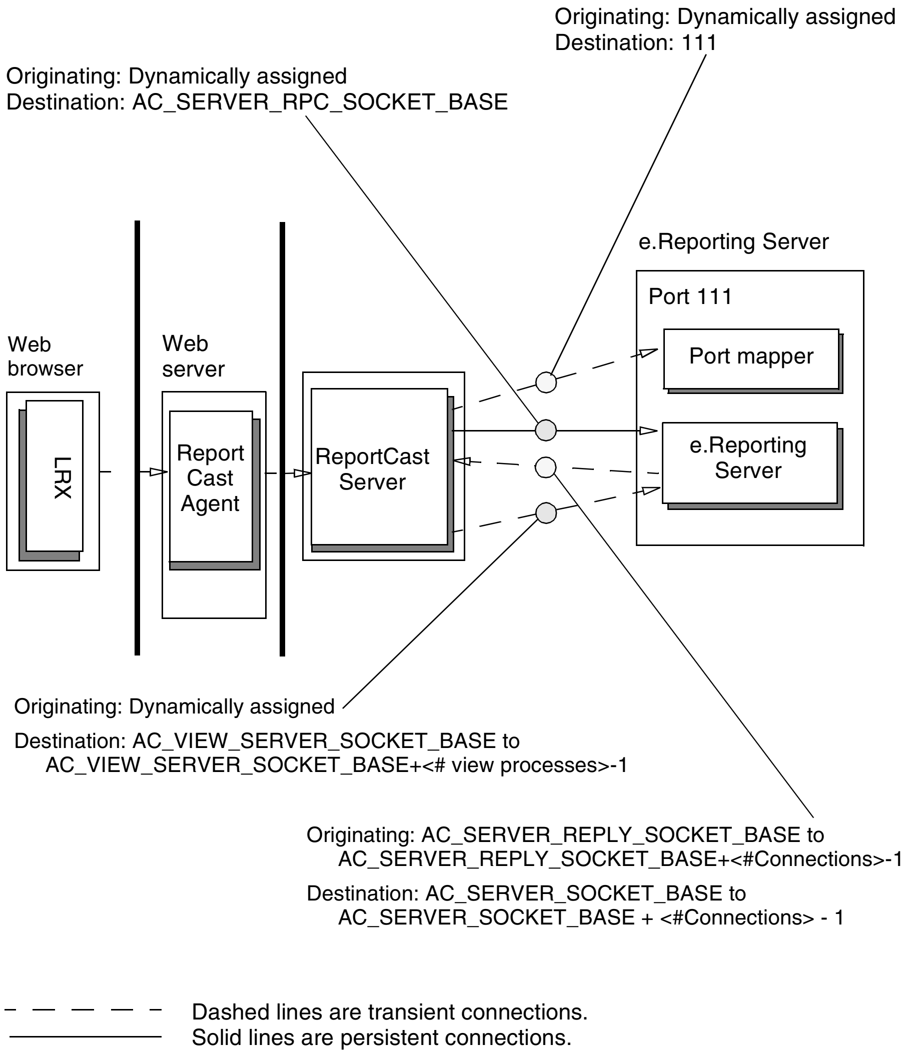
The following illustration shows a dual firewall configuration with the web server and ReportCastAgent in the DMZ.



Web browsers communicate with the web server through one well-known port, usually port 80, in the outer firewall. ReportCast Agent forwards requests through the inner firewall to the corporate net and ReportCast Server. ReportCast Server processes the requests and passes them to the e.Reporting Server using TCP/IP. The e.Reporting Server communicates with the Report Encyclopedia and databases using TCP/IP.

This dual firewall option provides an effective screen against unauthorized access. Internet traffic comes through only one port, generally using only one protocol: HTTP or HTTPS. Users on the Internet cannot gain direct access to the corporate network, because its ports and IP addresses are known only internally. Users on the Internet have limited options for accessing the web server machine, because they can only communicate using one protocol. For example, they cannot access the corporate network using FTP or Telnet.

The following illustration shows the inner firewall configuration.


Originating: Dynamically assigned
Destination: 111
Originating: Dynamically assigned
Destination: AC_SERVER_RPC_SOCKET_BASE
e.Reporting Server
Port 111
Port mapper
e.Reporting Server
Web browser
LRX
Web server
Report Cast Agent
ReportCast Server
Originating: Dynamically assigned
Destination: AC_VIEW_SERVER_SOCKET_BASE to AC_VIEW_SERVER_SOCKET_BASE+<# view processes>-1
Originating: AC_SERVER_REPLY_SOCKET_BASE to AC_SERVER_REPLY_SOCKET_BASE+<#Connections>-1
Destination: AC_SERVER_SOCKET_BASE to AC_SERVER_SOCKET_BASE + <#Connections> - 1
Dashed lines are transient connections.
Solid lines are persistent connections.

For information about configuring the originating and destination ports, see “Configuring the e.Reporting Server port,” “Configuring data reply ports,” and “Configuring the view process port,” earlier in this chapter.

NAT considerations for dual firewalls with ReportCast Agent in the DMZ

Actuate supports the NAT Static Address Translation algorithm, where users establish one-to-one connections between corporate (inside) network addresses and global (outside, Internet) network addresses. Actuate components initiate connections from either side of the firewall or routers. The IP address mapping must operate and be consistent from both sides of the firewall or routers.

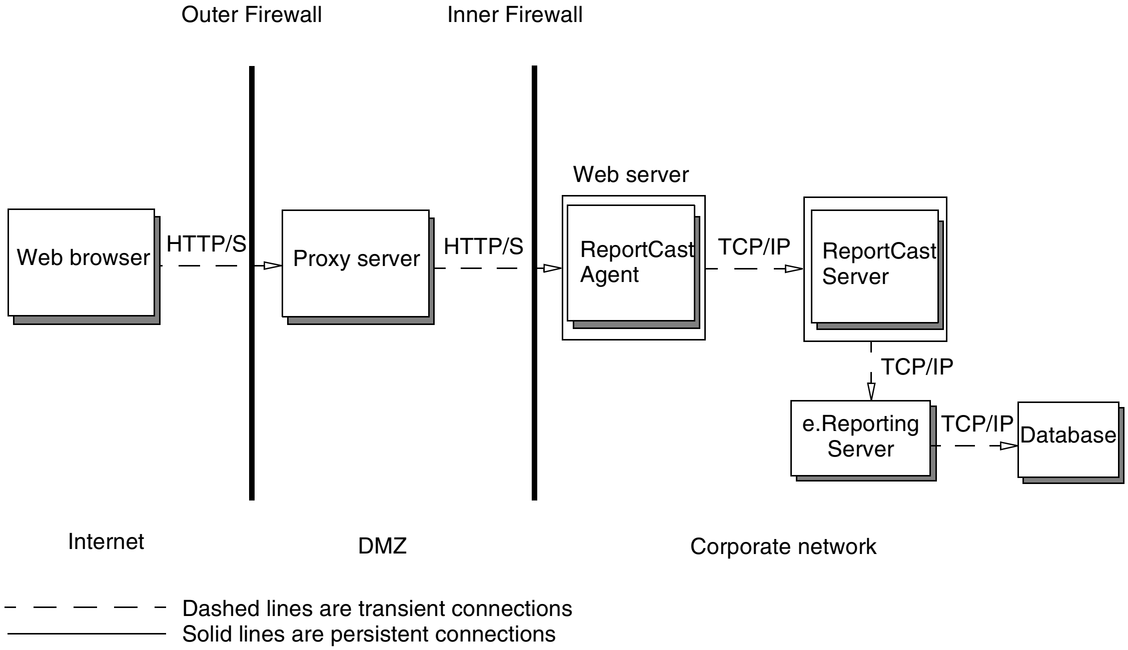
This configuration does not support the following NAT algorithms:

- Dynamic Source Address Translation
- Port Address Translation (PAT)
- Destination Address Rotary Translation

Dual firewalls with a proxy server

In this configuration, a proxy server resides in the DMZ. The web server, ReportCast Agent, ReportCast Server, and e.Reporting Server reside behind the inner firewall on the corporate network. The proxy server accepts HTTP or HTTPS traffic from the Internet, then passes the HTTP or HTTPS requests to the web server on the corporate network. ReportCast receives requests from the web server and passes the requests to the e.Reporting Server using TCP/IP.

The following illustration shows a dual firewall configuration with a proxy server in the DMZ.



The proxy server handles all communications with the Internet using HTTP through one well-known port, usually port 80. The proxy server communicates with the web server on the corporate network using HTTP through one internally-known port. ReportCast and the e.Reporting Server use TCP/IP to communicate.

This configuration option provides one access point for users on the Internet, through one IP address and well-known port, and one access point to the corporate network. The proxy configuration severely limits the possibilities of accessing the corporate network from the Internet.

NAT considerations for dual firewalls with a proxy server

Configure NAT as you usually would for use with a proxy server. Refer to your NAT and proxy server documentation for configuration information.

NAT considerations for the split ReportCast configuration

There are no special NAT considerations for the split ReportCast configuration.

Placing the e. Reporting Server in the DMZ

This configuration is possible, but not recommended. In this configuration, the web server, ReportCast, and the e.Reporting Server reside in the DMZ. The e.Reporting Server contains corporate data in the form of reports. Corporate data is vulnerable to outside attack in this configuration, which presents an unacceptable security risk.

Index